Applied Soft Computing xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

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

A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding

Bahriye Akay*

Erciyes University, The Department of Computer Engineering, 38039 Melikgazi, Kayseri, Turkiye

ARTICLE INFO

Article history:
Received 12 September 2011
Received in revised form 16 February 2012
Accepted 26 March 2012
Available online xxx

Keywords:
Image segmentation
Multilevel thresholding
Kapur's entropy
Between-class variance
Artificial bee colony
Particle swarm optimization

ABSTRACT

Segmentation is a critical task in image processing. Bi-level segmentation involves dividing the whole image into partitions based on a threshold value, whereas multilevel segmentation involves multiple threshold values. A successful segmentation assigns proper threshold values to optimise a criterion such as entropy or between-class variance. High computational cost and inefficiency of an exhaustive search for the optimal thresholds leads to the use of global search heuristics to set the optimal thresholds. An emerging area in global heuristics is swarm-intelligence, which models the collective behaviour of the organisms. In this paper, two successful swarm-intelligence-based global optimisation algorithms, particle swarm optimisation (PSO) and artificial bee colony (ABC), have been employed to find the optimal multilevel thresholds. Kapur's entropy, one of the maximum entropy techniques, and between-class variance have been investigated as fitness functions. Experiments have been performed on test images using various numbers of thresholds. The results were assessed using statistical tools and suggest that Otsu's technique, PSO and ABC show equal performance when the number of thresholds is two, while the ABC algorithm performs better than PSO and Otsu's technique when the number of thresholds is greater than two. Experiments based on Kapur's entropy indicate that the ABC algorithm can be efficiently used in multilevel thresholding. Moreover, segmentation methods are required to have a minimum running time in addition to high performance. Therefore, the CPU times of ABC and PSO have been investigated to check their validity in real-time. The CPU time results show that the algorithms are scalable and that the running times of the algorithms seem to grow at a linear rate as the problem size increases.

1. Introduction

Segmentation is an important and elementary step in image processing. It involves partitioning an image into non-overlapping, homogenous regions containing similar objects. Methods for evaluating the performance of a segmentation method fall into two categories depending on whether a ground-truth reference image is required: supervised and unsupervised evaluation, respectively. Unsupervised methods are preferable in real-time applications because they do not require a manually segmented image. If the image is split into two classes, such as the background and the object of interest, it is called bi-level thresholding. Bi-level thresholding is extended to multilevel thresholding to obtain more than two classes.

The thresholds can be derived at a local or global level [11]. In local thresholding, a different threshold is assigned for each part of the image, while in global thresholding, a single global threshold value is set to the whole image. At the local or global level, the probability density function of the grey level histogram can be handled by a parametric or a nonparametric approach to find the thresholds. In the parametric approaches, the statistical parameters of the classes in the image are estimated. They are computationally expensive, and their performance may vary depending on the initial conditions. In the nonparametric approaches, the thresholds are determined by maximising some criteria, such as between-class variance [12] or entropy measures.

In the literature, there are many entropy measures, such as minimum cross entropy, maximum entropy and variants of each. Minimum cross entropy (mutual information content) quantifies the average information for discrimination between two independent distributions [13]. This requires that probability distributions for both the original image and the segmented image are known. When one of the distributions is assumed to be uniform, it becomes the maximisation of the entropy. The maximum entropy principle is a special case of the general cross-entropy minimisation principle [13]. The complexity of the minimum cross entropy is higher than the maximum entropy. Kapur's entropy is a generalised form of Shannon's entropy for discrete random variables. Although there are some different entropy measures for maximum entropy, only Shannon and Kapur's entropy measures always lead to positive

* Tel.: +90 352 437 49 01x32578; fax: +90 352 437 57 84.
E-mail address: bahriye@erciyes.edu.tr

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.03.072

probabilities and a global maximum for the entropy [14, p. 28]. Sezgin and Sankur note that Kapur's entropy has produced better average scores than the other entropy-based measures on non-destructive testing images [15]. Therefore, in this study, Kapur's entropy has been chosen as the entropy measure due to its feasibility for segmentation.

Heuristic methods for finding optimal thresholds gained the attention of researchers because of the computational inefficiency of the traditional exhaustive methods [16]. Most of the related works were conducted using swarm algorithms, including ant colony optimisation (ACO) [17], PSO [18–21], bacterial foraging (BF) [10,22,23], honey bee mating optimisation (HBMO) [16] and ABC [24,25]. These methods use different versions of the basic algorithms and different objective functions, such as maximum entropy, Kapur's entropy, and between-class variance.

The PSO algorithm [26], introduced by Eberhart and Kennedy in 1995, is a swarm-based stochastic optimisation technique that models the social behaviour of bird flocking or fish schooling. It is well adapted to the optimisation of nonlinear functions in multidimensional space. PSO has received significant interest from researchers in several different research areas [27–32].

The ABC algorithm [33] developed in 2005 by Karaboga mimics the foraging behaviour of honey bees. The ABC algorithm has shown superior performance on numerical optimisation [34,35] and has been applied to many problems encountered in different research areas [36–40].

Both the PSO and ABC algorithms are simple but efficient models, and their search cycles are easy to implement compared to many other search heuristics. PSO and especially ABC have a few pre-defined control parameters that must be set. They have different selection and search operators, and they conduct exploration at different levels. They appear to have high performance in numeric optimisation [41]. One purpose of this paper is to investigate the search abilities of PSO and ABC algorithms on multi-level thresholding. In addition, we investigate the fitness functions and present a comprehensive comparative study of the two algorithms used as unsupervised-nonparametric-global-multilevel thresholding methods. Kapur's entropy has been compared to the between-class variance based on two fidelity criteria: the peak-to-signal-noise (PSNR) ratio and the structural similarity index (SSIM). This is the first study which applies the ABC algorithm to multi level thresholding based on the between-class variance and which investigates the effect of fitness criteria on the performance of the algorithms.

The performance of the algorithms has been evaluated on several images. The statistical analysis of the results suggests that the ABC algorithm is more efficient and more robust in multilevel thresholding, while the PSO algorithm tends to produce unsatisfactory results and show unstable behaviour. Images segmented using Kapur's entropy as the fitness function seems to have higher fidelity than those using between-class variance. However, when the number of thresholds is two, the between-class variance fitness function seems to be better than or similar to Kapur's entropy. A comparison based on CPU times has been presented because the running time is critical in real-time applications. The CPU times of the algorithms are reasonable and grow at linear rate as the problem dimension increases.

The remainder of the paper is organised as follows. In the second section, the thresholding problem is formulated, and then, Kapur's entropy and the between-class variance are addressed. In the next section, a brief overview of swarm intelligence, particle swarm optimisation, and artificial bee colony algorithms are provided. Section 4 presents the application of the algorithms to multilevel thresholding, reports the experimental results, and provides some discussions. Finally, Section 5 is dedicated to the conclusion.

2. Formulation of the problem

Thresholding an image at a threshold t partitions the image I into disjoint subsets based on the value of t. Bi-level thresholding assigns pixels with the grey level below t to a group, M_0, and assigns those pixels with the grey level above t to another group, M_1. At the end of the thresholding, a binary image is constructed that includes two groups. Assume that an image can be represented by L gray levels, bi-level thresholding can be defined as in (1):

$$\begin{aligned} M_0 &= \{g(x,y) \in I \mid 0 \le g(x,y) \le t-1\} \\ M_1 &= \{g(x,y) \in I \mid t \le g(x,y) \le L-1\} \end{aligned} \tag{1}$$

Multilevel thresholding uses more than one threshold value and creates an output image with multiple groups (2):

$$\begin{aligned} M_0 &= \{g(x,y) \in I \mid 0 \le g(x,y) \le t_1 - 1\} \\ M_1 &= \{g(x,y) \in I \mid t_1 \le g(x,y) \le t_2 - 1\} \\ M_i &= \{g(x,y) \in I \mid t_i \le g(x,y) \le t_{i+1} - 1\} \\ M_m &= \{g(x,y) \in I \mid t_m \le g(x,y) \le L-1\} \end{aligned} \tag{2}$$

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

It can be easy to obtain the threshold for bi-level thresholding. However, computing the thresholds for the multilevel thresholding can require high computational efforts. In the nonparametric approaches, thresholds are assigned to ensure that that the histogram of the thresholded image satisfies the desired criteria based on the between-class variance, as in the Otsu technique [42], or based on an entropy criterion, such as Kapur's entropy [43].

2.1. Based on Kapur's entropy

An efficient segmentation technique is entropy-based thresholding based on the probability distribution of the gray level histogram. The entropy is maximum when the optimal thresholds separating the classes are assigned properly. Therefore, the purpose is to find the optimal thresholds yielding the maximum entropy. The entropy of a discrete source is often obtained from the probability distribution $p = p_i$, where p_i is the probability of the system in possible state i [44]. The probability of each gray level i is the relative occurrence frequency of the gray level i, normalized by the total number of gray levels (3):

$$p_i = \frac{h(i)}{\sum_{i=0}^{L-1} h(i)}, \quad i = 0, \ldots, L-1 \tag{3}$$

Kapur's entropy measures the compactness and separability of classes. For bi-level thresholding, Kapur's entropy may be described by (4):

$$\begin{aligned} H_0 &= -\sum_{i=0}^{t-1} \frac{p_i}{\omega_0} \ln \frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{t-1} p_i \\ H_1 &= -\sum_{i=t}^{L-1} \frac{p_i}{\omega_1} \ln \frac{p_i}{\omega_1}, \quad \omega_1 = \sum_{i=t}^{L-1} p_i \end{aligned} \tag{4}$$

The threshold is optimum when the summation of the class entropies is maximum (5).

$$t^* = arg\ \max(H_0 + H_1) \tag{5}$$

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Kapur's entropy can be extended for multilevel thresholding [22] as given by (6):

$$\begin{aligned}
H_0 &= -\sum_{i=0}^{t_1-1}\frac{p_i}{\omega_0}\ln\frac{p_i}{\omega_0}, \quad \omega_0 = \sum_{i=0}^{t_1-1}p_i \\
H_1 &= -\sum_{i=t_1}^{t_2-1}\frac{p_i}{\omega_1}\ln\frac{p_i}{\omega_1}, \quad \omega_1 = \sum_{i=t_1}^{t_2-1}p_i \\
H_2 &= -\sum_{i=t_2}^{t_3-1}\frac{p_i}{\omega_2}\ln\frac{p_i}{\omega_2}, \quad \omega_2 = \sum_{i=t_2}^{t_3-1}p_i \\
H_j &= -\sum_{i=t_j}^{i=t_{j+1}-1}\frac{p_i}{\omega_j}\ln\frac{p_i}{\omega_j}, \quad \omega_j = \sum_{i=t_j}^{i=t_{j+1}-1}p_i \\
H_m &= -\sum_{i=t_m}^{L-1}\frac{p_i}{\omega_m}\ln\frac{p_i}{\omega_m}, \quad \omega_m = \sum_{i=t_m}^{L-1}p_i
\end{aligned} \tag{6}$$

The m-tuple optimal thresholds are obtained by (7) that tries to maximise the objective function:

$$(\vec{t})^* = arg\ \max\left(\sum_{i=0}^{m}H_i\right) \tag{7}$$

2.2. Based on between-class variance

Thresholding based on the between-class variance is another nonparametric segmentation method that divides the whole image into classes so that the variance of the different classes is maximum.

Otsu [42] defined the between-class variance as the sum of sigma functions of each class by (8)

$$f(t) = \sigma_0 + \sigma_1 \tag{8}$$

$$\begin{aligned}
\sigma_0 &= \omega_0(\mu_0 - \mu_T)^2 \\
\sigma_1 &= \omega_1(\mu_1 - \mu_T)^2
\end{aligned} \tag{9}$$

where μ_T is the mean intensity of the original image. In bi-level thresholding, the mean level of each class, (μ_i), can be calculated by (10):

$$\mu_0 = \sum_{i=0}^{t-1}\frac{ip_i}{\omega_0}, \quad \mu_1 = \sum_{i=t}^{L-1}\frac{ip_i}{\omega_1} \tag{10}$$

The optimal threshold can be derived by maximising the between-class variance function, (11):

$$t^* = arg\ \max(f(t)) \tag{11}$$

Bi-level thresholding based on the between-class variance can be extended to multi-level thresholding. The extended between-class variance function is calculated by (12):

$$f(t) = \sum_{i=0}^{m}\sigma_i \tag{12}$$

The sigma functions are calculated by (13) and the mean levels are calculated by (14):

$$\begin{aligned}
\sigma_0 &= \omega_0(\mu_0 - \mu_T)^2 \\
\sigma_1 &= \omega_1(\mu_1 - \mu_T)^2 \\
\sigma_2 &= \omega_2(\mu_2 - \mu_T)^2 \\
\sigma_j &= \omega_j(\mu_j - \mu_T)^2 \\
\sigma_m &= \omega_m(\mu_m - \mu_T)^2
\end{aligned} \tag{13}$$

$$\mu_0 = \sum_{i=0}^{t_1-1}\frac{ip_i}{\omega_i}, \quad \mu_1 = \sum_{i=t_1}^{t_2-1}\frac{ip_i}{\omega_i}, \quad \mu_2 = \sum_{i=t_2}^{t_3-1}\frac{ip_i}{\omega_i}, \quad \mu_j = \sum_{i=t_j}^{t_{j+1}-1}\frac{ip_i}{\omega_i},$$

$$\mu_m = \sum_{i=t_m}^{L-1}\frac{ip_i}{\omega_i} \tag{14}$$

The optimal thresholds are found by maximising the between-class variance function (15):

$$(\vec{t})^* = arg\ \max\left(\sum_{i=0}^{m}\sigma_i\right) \tag{15}$$

3. Brief explanations of the algorithms in the study

In the literature, some swarm algorithms simulating the swarm behaviour of the organisms in nature have been presented and have been successfully applied to a large number of problems in the literature [48,36].

Although there are some other swarm intelligence-based algorithms, in this study PSO and ABC algorithms have been used because they employ less control parameters compared to others in the literature. This is important because tuning the control parameters of an algorithm might be more difficult than the problem itself. Because multilevel thresholding is not a large-scale problem, the algorithm chosen is advised to employ a little number of control parameters. Another issue related with an algorithm is that it should have low complexity in addition to high performance. Both ABC and PSO algorithms are efficient tools to solve numeric problems in global optimization [41,26]. Moreover, they are based on simple models and their computation times are low, which make them preferable in real-time implementation of signal processing applications.

A brief description of the PSO and the ABC algorithms are given in the following subsections.

3.1. Particle swarm optimisation

The PSO algorithm [26], introduced by Eberhart and Kennedy in 1995, models the social behaviour of bird flocking or fish schooling. In PSO, a swarm is a collection of the particles moving in search space. The particles change their positions depending on their previous experience and the best experience of the swarm to find the global optimum.

All of the particles are initialised at random positions by (16), and they start to move in the search space by changing their velocities and then positions:

$$x_{ij} = x_j^{min} + \text{rand}(0,1)(x_j^{max} - x_j^{min}) \tag{16}$$

where $\vec{x}_i$ is the position of ith particle, $i = 1, \ldots, SS$, SS is swarm size, $j = 1, \ldots, D$ and D is the dimension of the problem. Once a population is generated, the algorithm starts to iterate the steps as given in Algorithm 1.

Algorithm 1 *(Main steps of the PSO algorithm).*

```
Initialize the population
repeat
    Calculate the fitness values of the particles
    Update the best experience of each particle
    Choose the best particle
    Calculate the velocities of the particles
    Update the positions of the particles
until requirements are met
```

In the loop of the PSO algorithm, each particle's position is evaluated in the cost function, and a fitness value is assigned to each. The best-so-far position of the population (global best, $\vec{g}(t)$) is kept

in memory to be used in velocity calculations. The best-so-far position of the particle ($\vec{p}(t)$) is found. The velocity of each particle is updated by (17) each time step t:

$$\vec{v}(t+1) = \omega\vec{v}(t) + \phi_1 \text{rand}(0,1)(\vec{p}(t) - \vec{x}(t)) + \phi_2 \text{rand}(0,1)(\vec{g}(t) - \vec{x}(t)) \quad (17)$$

The difference of ($\vec{p}(t)$) and the current position, and the difference of ($\vec{g}(t)$) and the current position are weighted by random terms and control parameters ϕ_1 and ϕ_2, called cognitive and social components. Therefore, the particles tend to move to promising areas in the search space. The parameter ω is called the inertia weight and controls the magnitude of the old velocity, $\vec{v}(t)$.

Furthermore, v_i at any time step of the algorithm is constrained by the parameter v_{max}. If the sum of accelerations would cause the velocity on a dimension to exceed a user-defined parameter, v_{max}, then the velocity on that dimension is limited to v_{max}.

The new position of each particle is derived by adding the new velocity to the previous position (18):

$$\vec{x}(t+1) = \vec{x}(t) + \vec{v}(t+1) \quad (18)$$

After a fitness value is assigned to the new solution, a particle does not consider whether the new position ($x(t+1)$) is better or not than the previous one ($x(t)$). However, the fitness value is used to update the best-so-far position of the particle and the best-so-far position of the population.

Detailed pseudocode of the PSO algorithm is given in Algorithm 2.

Algorithm 2 *(The PSO algorithm).*

Data: Set the control parameters of the PSO algorithm.
SS: Swarm Size,
ω: Inertia,
ϕ_1: Cognitive Component,
ϕ_2: Social Component,
v_{max}: Upper bound for velocity
MCN: Maximum iteration Number.
begin
  //Initialization;
  for $i = 1$ *to* SS **do**
    $\vec{X}(i) \longleftarrow$ random solution by Eq. 16 ;
    $\vec{v}(i) \longleftarrow$random;
    $\vec{P}(i) \longleftarrow \vec{X}(i)$;
    $f(i) \longleftarrow f(X(i))$;
    $Pf(i) \longleftarrow f(i)$;
  end
  $iter = 1$;
  while $iter < MCN$ **do**
    $best = \{i : Pf(i) = min(Pf)\}$;
    for $i = 1$ *to* SS **do**
      $\vec{v}(i) = \omega\vec{v}(i) + \phi_1 \text{rand}(0,1)(\vec{P}(i) - \vec{X}(i)) + \phi_2 \text{rand}(0,1)(\vec{P}(best) - \vec{X}(i))$;
      if $v(i) > v_{max}$ **then**
        $v(i) = v_{max}$;
      end
      $\vec{X}(i) = \vec{X}(i) + \vec{v}(i)$;
      $f(i) \longleftarrow f(\vec{X}(i))$ evaluate new solution;
      if $f(i) < Pf(i)$ **then**
        $P(i) \longleftarrow X(i)$;
        $Pf(i) \longleftarrow f(i)$;
      end
    end
    $iter++$;
  end
end

3.2. Artificial bee colony algorithm

The artificial bee colony algorithm [33] developed by Karaboga in 2005 is a global optimisation algorithm that mimics the foraging behaviour of honey bees. In nature, there is a division of labour in the hive, and the forager bees work collectively without a central control mechanism to maximise the amount of nectar loaded into

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

the hive. The behaviour of real bees and a detailed analogy can be found in [33].

In the algorithm, each food source position corresponds to a solution. Each bee selects a source and searches the neighbourhood of the solution (exploitation). There are three types of bees in the foraging process: employed bees, onlooker bees and scout bees. The bees are classified according to how they select the food source to exploit. An employed bee searches the neighbourhood of the source in her memory and updates her memory if she finds a better solution; otherwise, she counts the number of the searches around the source in her memory. An onlooker bee does not have any source in her memory and selects a probably profitable food source (recruitment). Information about the profitability of the sources is gathered from the experiences of the employed bees (interaction, communication). When an onlooker bee chooses a source, she searches the neighbourhood of the source. She updates the food source position when she finds a better solution as an employed bee does.

If the number of the searches associated with a source exceeds "*limit*", it means that the solution has been sufficiently exploited, and it may be assumed to be exhausted. The bee of the exhausted source discards her source (abandonment) and becomes a scout. A scout bee selects a random source (exploration) to exploit. The ABC algorithm models the exploration, exploitation, recruitment and abandonment behaviours in the foraging process of honey bees.

The main phases of the algorithm are given step-by-step in Algorithm 3.

Algorithm 3 *(Main steps of the ABC algorithm).*

```
Initialization
Evaluation
repeat
    Employed Bee Phase
    Onlooker Bee Phase
    Scout Bee Phase
    Memorize the best solution achieved so far
until A termination criteria is satisfied
```

In the initialisation phase, a food source population is generated randomly by (16) as in the PSO algorithm.

In the employed bee phase, a search is conducted by (19) around the source in each bee's memory. If the solution produced by (19) is better than the solution in the bee's memory, the memory is updated by a greedy selection approach:

$$x'_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{19}$$

where $k \in [1, CS]$ is a uniform random index, CS is the number of food sources, $j \in [1, D]$ is uniform random index and D is the dimension of the problem.

The employed bees share their experiences with the onlookers, which is fulfilled by assigning each solution a normalised fitness-based probability (20):

$$p_i = \frac{fitness_i}{\sum_{i=1}^{CS} fitness_i} \tag{20}$$

Each onlooker bee chooses a source by a probabilistic, roulette-wheel-like selection. How this selection scheme is applied can be seen in the detailed Algorithm 4. After choosing a source, a local search is carried out by (19), and a greedy selection is applied to keep the solution that is better for the population.

In both the employed bee and the onlooker bee phases, if a solution cannot be improved by local searches during a number of cycles ("*limit*"), that source is assumed to be exhausted, and its bee becomes a scout in the scout bees' phase. A scout bee finds a new random solution produced by (16) to be replaced with the exhausted source.

A detailed pseudocode of the ABC algorithm is given in Algorithm 4.

Algorithm 4 *(The ABC algorithm).*

```
Data: Set the control parameters of the ABC algorithm.
CS: Number of Foods,
MCN: Maximum Cycle Number,
limit: Maximum number of trial for abandoning a source
begin
    //Initialization;
    for s = 1 to CS do
        X(s) ⟵ random solution by Eq. 16; f_s ⟵ f(X(s)); trial(s) ⟵ ∅;
    end
    cycle = 1;
    while cycle < MCN do
        //Employed Bees' Phase;
        for s = 1 to CS do
            x' ⟵ a new solution produced by Eq. 19;
            f(x') ⟵ evaluate new solution;
            if f(x') < f_s then
                X(s) ⟵ x'; f_s ⟵ f(x'); trial(s) ⟵ ∅;
            else
                trial(s) = trial(s) + 1;
            end
        end
        //Calculate probabilities for onlookers by Eq. 20;
        //Onlooker Bees' Phase;
        s ⟵ 0 ;
        t ⟵ 0 ;
        while t < CS do
            r ⟵ rand(0, 1);
            //Probabilistic Selection;
            if r < p(s) then
                t ⟵ t + 1;
                x' ⟵ a new solution produced by Eq. 19;
                f(x') ⟵ evaluate new solution;
                if f(x') < f_s then
                    X(s) ⟵ x'; f_s ⟵ f(x'); trial(s) ⟵ ∅;
                else
                    trial(s) = trial(s) + 1;
                end
            end
            s ⟵ (s + 1) mod (CS − 1);
        end
        //Scout bee phase;
        mi = {s : trial(s) = max(trial)};
        if trial(mi) > limit then
            X(mi) ⟵ random solution by Eq. 16;
            f_mi = f(X(mi));
            trial(mi) ⟵ ∅;
        end
        cycle + +;
    end
```

4. Experiments and results

4.1. Experimental setup

The multilevel thresholding problem deals with finding optimal thresholds within the range $[0, L-1]$ that maximise a fitness criterion. The dimension of the optimisation problem is the number of thresholds (m), and the search space is $[0, L-1]^m$. In this study, optimal multilevel thresholding has been carried out by an unsupervised global-level nonparametric approach. In the approach, PSO and ABC algorithms have been employed separately, and their search efficiencies have been compared. Two different fitness criteria have been examined: Kapur's entropy and between-class variance.

In the algorithms, each candidate solution corresponds to a m-element set of threshold values to be optimised. In the initialisation phases of the algorithms, a population of the solutions are generated at random within the range $[0, L-1]$ on each dimension. Once a population is generated, solutions in the population are assigned a fitness value. Then, the algorithms start to search the optimal threshold values by their selection and perturbation operators.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

In the PSO algorithm, particle positions represent the thresholds, and the aim is to find the optimal thresholds by changing the velocities and the positions of particles in the search space by (17) and (18). The PSO algorithm iterates the steps described in Section 3.1 until a termination criteria is satisfied.

In the ABC algorithm, food source positions correspond to the thresholds, and the algorithm iterates the employed bee, the onlooker bee and the scout bee phases to select some sources and to execute the search operators by (19). How the solutions are selected and perturbed in each phase is described in Section 3.2.

Both of the algorithms have been used in their standard versions. The population size (*SS*) in the PSO algorithm was set to 40. For fair comparison, the number of food sources, *CS*, has been set to 20 in the ABC algorithm to make the number of function evaluations the same as that of the PSO algorithm; the ABC algorithm evaluates the solutions both in the employed bee and onlooker bee phases. Both algorithms were terminated at 100 iterations. The other control parameter of the ABC algorithm, *limit*, was set to 50. For the PSO algorithm, ω was set to 0.6, both ϕ_1 and ϕ_2 were set to 2.

Experiments were carried out on some test images, shown in Fig. 1, so that the classes after segmentation could be easily seen. After the test images were evaluated, experiments were repeated on twelve real images, shown in Fig. 2, to illustrate the validation of the ABC and PSO algorithms. Two of the images are well-known test images called Lena and Cameraman. The other ten images, Butterfly, Cloud, Elephants, Landscape, Ostrich, Plane, Snake, Starfish, Wherry and Zebra, were taken from the Berkeley Segmentation Dataset [49]. Each image has a unique grey level histogram. Most of the images are difficult to segment due to the multimodality of the histograms.

The number of thresholds (m) investigated in the experiments were 2–5. The algorithms have randomised characteristics; therefore, all of the experiments were repeated 30 times for each image and for each m value.

While evaluating results, peak-to-signal ratio (PSNR) and structural similarity indices were also reported, in addition to values of the fitness functions, thresholds and segmented images. PSNR (21) gives the similarity of an image against a reference image based on the mean square error (MSE) of each pixel:

$$PSNR(x,y) = 20\log_{10}\left(\frac{255}{\sqrt{MSE(x,y)}}\right) \tag{21}$$

SSIM given by Eq. (22) evaluates the visual similarity between the original image and the reconstructed image. The SSIM index

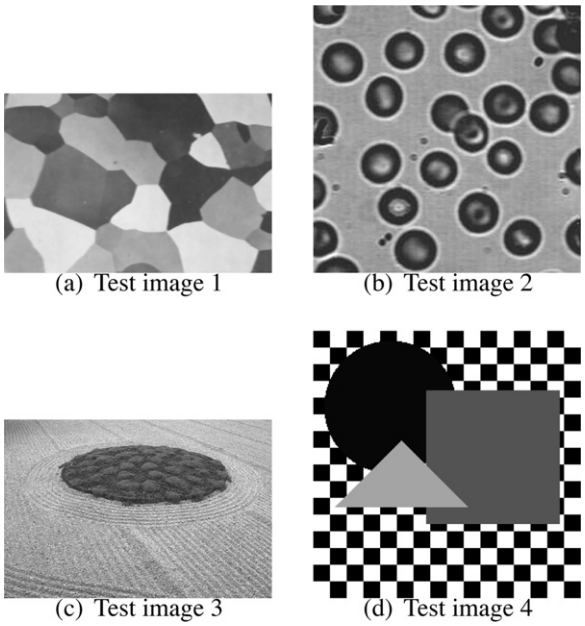


Fig. 1. Test images used in the experiments. (a) Test image 1, (b) test image 2, (c) test image 3, and (d) test image 4.

combines luminance comparison, contrast comparison and structure comparison and satisfies symmetry, boundedness and unique maximum properties [50]:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{22}$$

where μ_x is the average of x, μ_y is the average of y, σ_x^2 is the variance of x, σ_y^2 is the variance of y,σ_{xy} is the covariance of x and y, $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ stabilise the division with weak denominator, L is the dynamic range of the pixel-values, k_1=0.01 and k_2=0.03.

4.2. Experiment 1: Minimising between-class variance

In the first part of the experiments, the between-class variance fitness criterion was maximised to construct an optimally segmented image. Both the PSO and the ABC algorithms used the same criterion. The results of Otsu's method have also been reported to provide some comparative experiments with an existing multilevel thresholding method. Table 1 presents the threshold values and the corresponding between-class variance values of the best solutions produced by the methods after 30 runs. Table 2

Table 1
Comparison of the best thresholds obtained from the methods based on the between-class variance criterion for test images.

Image	m	Between-class variance			Thresholds		
		Otsu	PSO	ABC	Otsu	PSO	ABC
Test 1	2	**0.919368**	**0.919368**	**0.919368**	127, 192	127, 192	127, 192
	3	0.964925	0.964417	**0.964979**	49, 117, 192	52, 111, 194	51, 117, 193
	4	0.976208	0.975534	**0.976242**	49, 111, 169, 210	54, 114, 176, 214	49, 113, 168, 210
	5	0.982812	0.981096	**0.983081**	44, 80, 123, 170, 211	48, 73, 117, 163, 203	39, 73, 120, 169, 211
Test 2	2	**0.916317**	**0.916317**	**0.916317**	73, 138	73, 138	73, 138
	3	0.951716	0.960978	**0.961303**	54, 112, 159	27, 94, 153	27, 98, 159
	4	0.963016	0.972280	**0.973595**	48, 104, 153, 179	12, 61, 98, 159	12, 61, 112, 159
	5	0.974855	0.978225	**0.981485**	33, 94, 133, 159, 183	11, 48, 98, 153, 183	12, 61, 104, 139, 165
Test 3	2	**0.854607**	**0.854607**	**0.854607**	117, 172	117, 172	117, 172
	3	0.911062	0.910808	**0.911151**	92, 140, 179	91, 138, 180	91, 140, 180
	4	0.940590	0.933302	**0.940639**	78, 120, 157, 189	88, 117, 146, 180	78, 120, 156, 189
	5	0.952902	0.952766	**0.957587**	53, 91, 127, 161, 190	75, 94, 132, 163, 193	72, 109, 141, 168, 196
Test 4	2	**0.939148**	**0.939148**	**0.939148**	0, 164	0, 164	0, 164
	3	0.958182	**0.979232**	**0.979232**	7, 82, 164	0, 7, 164	0, 7, 164
	4	**1.000000**	**1.000000**	**1.000000**	0, 7, 82, 164	0, 7, 82,164	0, 7, 82, 164

The values in bold face indicate the best result.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Fig. 2. Real images used in the experiments. (a) Butterfly image, (b) Cameraman image, (c) Cloud image, (d) Elephants image, (e) Landscape image, (f) Lena image, (g) Ostrich image, (h) Plane image, (i) Snake image, (j) Starfish image, (k) Wherry image, and (l) Zebra image.

gives the PSNR and SSIM values associated with the solutions in Table 1. The results indicate that all of the methods performed equally when the number of thresholds was 2, while the ABC algorithm is better than the two other methods in terms of the fitness criterion, the between-class variance. Tables 1 and 2 suggest that close variance values may not lead to similar PSNR values. The algorithms have not been given the PSNR value as a fitness value in the optimisation process; therefore, the algorithms are expected to minimise the fitness criterion, the between-class variance.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 2
PSNR and SSIM metrics of the test images segmented with the thresholds yielding the best between-class variance.

Image	m	PSNR (dB)			SSIM		
		Otsu	PSO	ABC	Otsu	PSO	ABC
Test 1	2	**24.1841934**	**24.184193**	**24.184193**	**0.914243**	**0.9142432**	**0.914243**
	3	**21.461313**	20.203420	16.534549	**0.891827**	0.879998	0.813326
	4	**26.083261**	21.405376	20.214157	**0.901105**	0.889645	0.879463
	5	**26.897088**	22.607626	18.364321	**0.908244**	0.859564	0.832477
Test 2	2	**24.648476**	**24.648476**	**24.648476**	**0.735714**	**0.7357142**	**0.735714**
	3	**24.388360**	23.895983	19.873400	0.718210	**0.737761**	0.608460
	4	**27.777715**	20.105586	20.105586	**0.818597**	0.708220	0.708220
	5	**29.335303**	17.937294	24.171421	**0.880487**	0.536128	0.710212
Test 3	2	**23.642603**	**23.642603**	**23.642603**	**0.738069**	**0.7380692**	**0.738069**
	3	**22.704023**	22.555665	19.027206	**0.648311**	0.637185	0.326372
	4	**24.884149**	23.042976	20.564769	**0.784643**	0.687478	0.653812
	5	**25.694946**	23.466148	24.863369	**0.826791**	0.740763	0.812234
Test 4	2	**17.324235**	**17.324235**	**17.324235**	**0.820157**	**0.8201572**	**0.820157**
	3	21.848488	**22.847084**	**22.847084**	**0.900168**	0.888762	0.888762
	4	Inf	Inf	Inf	**1.000000**	**1.000000**	**1.000000**

The values in bold face indicate the best result.

Table 3
Comparison of the best thresholds obtained from the methods based on the between-class variance criterion for real images.

Image	m	Between-class variance			Thresholds		
		Otsu	PSO	ABC	Otsu	PSO	ABC
Butterfly	2	**0.923426**	**0.923426**	**0.923426**	83, 155	83, 155	83, 155
	3	0.951457	0.950834	**0.951471**	72, 137, 196	72, 143, 198	71, 137, 195
	4	0.964544	0.963202	**0.965896**	59, 98, 150, 200	75, 117, 164, 197	66, 113, 161, 203
	5	0.975649	0.972272	**0.975807**	52, 81, 122, 168, 207	39, 74, 112, 154, 192	48, 79, 121, 166, 205
Cameraman	2	**0.939251**	**0.939251**	**0.939251**	69, 144	69, 144	69, 144
	3	0.958523	0.958158	**0.958647**	56, 118, 155	50, 109, 148	53, 115, 153
	4	0.970693	0.969825	**0.972792**	53, 112, 144, 170	16, 76, 137, 171	35, 90, 137, 168
	5	0.978398	0.978894	**0.980851**	40, 91, 134, 164, 199	18, 79, 121, 143, 168	29, 77, 118, 146, 171
Cloud	2	**0.910900**	**0.910900**	**0.910900**	94, 167	94, 167	94, 167
	3	0.945091	0.944911	**0.945211**	73, 134, 185	77, 136, 190	74, 134, 187
	4	0.956891	0.960631	**0.962228**	58, 100, 142, 192	38, 71, 136, 184	40, 81, 135, 188
	5	0.974485	0.973205	**0.974524**	38, 78, 120, 159, 200	43, 74, 123, 155, 202	38, 77, 120, 158, 200
Elephants	2	**0.914539**	**0.914539**	**0.914539**	141, 204	141, 204	141, 204
	3	0.934884	0.944093	**0.944904**	68, 129, 194	0.944093	115, 177, 215
	4	0.964154	0.960368	**0.964222**	70, 123, 177, 214	83, 117, 181, 220	71, 124, 178, 215
	5	0.969813	0.969674	**0.974094**	46, 80, 128, 179, 215	83, 117, 181, 220	70, 117, 163, 190, 220
Landscape	2	**0.913001**	**0.913001**	**0.913001**	53, 143	53, 143	53, 143
	3	0.970362	0.913001	**0.970678**	60, 128, 207	48, 123, 192	54, 125, 199
	4	0.984290	0.983053	**0.984402**	51, 102, 153, 204	39, 111, 152, 201	50, 103, 150, 209
	5	0.992186	0.989866	**0.992229**	49, 92, 121, 156, 212	38, 95, 123, 144, 203	48, 92, 122, 158, 210
Lena	2	**0.856599**	**0.856599**	**0.856599**	92, 151	92, 151	92, 151
	3	**0.929388**	0.928045	**0.929388**	64, 117, 168	61, 119, 174	64, 117, 168
	4	**0.957044**	0.955133	**0.957044**	57, 102, 138, 179	58, 100, 141, 188	57, 102, 138, 179
	5	0.968245	0.966593	**0.968394**	53, 95, 128, 153, 187	54, 93, 128, 147, 187	54, 96, 128, 155, 188
Ostrich	2	**0.968245**	**0.968245**	**0.968245**	74, 135	74, 135	74, 135
	3	0.900417	0.892161	**0.900453**	68, 101, 150	59, 91, 136	68, 100, 149
	4	0.931263	0.928326	**0.931586**	66, 94, 126, 178	64, 91, 116, 177	64, 91, 124, 176
	5	0.949724	0.943309	**0.951424**	60, 83, 103, 136, 186	50, 75, 105, 124, 189	55, 77, 99, 131, 179
Plane	2	**0.856388**	**0.856388**	**0.856388**	65, 118	65, 118	65, 118
	3	0.919361	0.918892	**0.920436**	76, 152, 180	92, 155, 180	85, 152, 180
	4	0.941907	0.942728	**0.949283**	51, 127, 161, 183	89, 127, 156, 185	75, 135, 163, 185
	5	0.957237	0.955297	**0.963686**	44, 96, 138, 165, 186	73, 120, 155, 172, 197	72, 128, 155, 173, 190
Snake	2	**0.803111**	**0.803111**	**0.803111**	86, 134	86, 134	86, 134
	3	0.883920	0.880593	**0.884028**	69, 106, 148	60, 104, 148	68, 107, 149
	4	0.923659	0.911147	**0.923684**	60, 93, 122, 160	51, 75, 112, 157	60, 94, 123, 161
	5	0.945009	0.939712	**0.945552**	58, 86, 111, 137, 175	56, 85, 101, 123, 165	55, 84, 108, 134, 171
Starfish	2	**0.843043**	**0.843043**	**0.843043**	84, 157	84, 157	84, 157
	3	0.920170	0.918130	**0.920181**	66, 117, 176	61, 118, 170	66, 117, 175
	4	0.948285	0.947286	**0.948576**	60, 101, 139, 187	60, 100, 143, 190	57, 99, 136, 185
	5	0.963940	0.960563	**0.964184**	52, 87, 117, 151, 195	49, 91, 118, 153, 183	49, 84, 115, 148, 193
Wherry	2	**0.890490**	**0.890490**	**0.890490**	107, 189	107, 189	107, 189
	3	0.952274	0.951487	**0.952338**	98, 157, 217	100, 160, 211	97, 158, 216
	4	0.967525	0.964610	**0.967565**	78, 117, 160, 218	89, 124, 164, 208	78, 117, 160, 216
	5	0.972141	0.973751	**0.976908**	49, 88, 124, 161, 218	71, 114, 144, 181, 237	76, 114, 150, 182, 223
Zebra	2	**0.827683**	**0.827683**	**0.827683**	99, 173	99, 173	99, 173
	3	0.905167	0.903116	**0.905790**	84, 121, 183	88, 120, 186	86, 125, 187
	4	0.939160	0.934875	**0.939189**	77, 105, 136, 195	80, 104, 134, 207	78, 107, 137, 196
	5	0.956017	0.951261	**0.956046**	70, 95, 118, 147, 202	78, 109, 137, 181, 224	71, 95, 118, 146, 201

The values in bold face indicate the best result.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 4
PSNR and SSIM metrics of the real images segmented with the thresholds yielding the best between-class variance.

Image	m	PSNR (dB)			SSIM		
		Otsu	PSO	ABC	Otsu	PSO	ABC
Butterfly	2	**23.061490**	**23.061490**	**23.061490**	**0.747907**	**0.7479072**	**0.747907**
	3	22.904615	**23.029528**	22.904630	0.741653	**0.745891**	0.741749
	4	24.051876	24.378128	**24.432183**	0.769665	0.786428	**0.789854**
	5	**25.476455**	24.895148	25.455667	**0.818003**	0.800623	0.817171
Cameraman	2	**24.397509**	**24.397509**	**24.397509**	**0.799569**	**0.799569**	**0.799569**
	3	**23.192950**	22.453228	22.944845	**0.821347**	0.816765	0.821248
	4	**25.831469**	25.418748	25.672221	0.856685	0.848919	**0.859049**
	5	**27.795392**	26.683982	27.131611	0.845344	**0.886308**	0.880184
Cloud	2	**23.076768**	**23.076768**	**23.076768**	**0.812141**	**0.812141**	**0.812141**
	3	22.341853	**22.475480**	22.341969	0.766146	**0.774645**	0.767287
	4	**23.364210**	23.219008	23.192089	**0.770485**	0.766276	0.765328
	5	**25.735851**	25.331309	25.663253	**0.826521**	0.818298	0.824951
Elephants	2	**24.427167**	**24.427167**	**24.427167**	**0.852178**	**0.852178**	**0.852178**
	3	21.351958	23.692038	**23.917239**	**0.882642**	0.863785	0.863881
	4	24.887706	24.949933	**24.971635**	0.866271	0.862851	**0.866289**
	5	25.402674	**27.762400**	26.596858	**0.872753**	0.843505	0.858627
Landscape	2	**21.810687**	**21.810687**	**21.810687**	**0.883460**	**0.883460**	**0.883460**
	3	**19.517602**	19.288271	19.417463	0.887010	0.890062	**0.890742**
	4	**23.204048**	23.103365	23.112415	0.887104	**0.889789**	0.882261
	5	23.845355	22.740831	**23.865987**	**0.912720**	0.890360	0.912383
Lena512	2	**22.966377**	**22.966377**	**22.966377**	**0.691397**	**0.691397**	**0.691397**
	3	22.000020	**22.143997**	22.000020	**0.716976**	0.716317	**0.716976**
	4	24.367972	**24.634869**	24.367972	0.752398	**0.753468**	0.752398
	5	26.572374	25.711971	**26.900380**	0.775978	0.772769	**0.780452**
Ostrich	2	**25.301596**	**25.301596**	**25.301596**	**0.846558**	**0.846558**	**0.846558**
	3	**23.313067**	22.277174	23.219015	**0.823085**	0.805828	0.820724
	4	**26.588903**	25.780154	26.479152	**0.859348**	0.850671	0.858778
	5	**27.919597**	27.044707	27.699910	**0.874874**	0.866096	0.873920
Plane	2	**28.348562**	**28.348562**	**28.348562**	**0.907461**	**0.907461**	**0.907461**
	3	27.918494	**28.073372**	27.947954	0.923213	0.921294	**0.923360**
	4	29.257115	28.931820	**29.861264**	0.914746	**0.924851**	0.915546
	5	30.396679	31.482925	**31.840694**	**0.918520**	0.917715	0.918071
Snake	2	**23.748996**	**23.748996**	**23.748996**	**0.750523**	**0.750523**	**0.750523**
	3	22.915673	22.705653	**22.993334**	0.741581	0.729620	**0.743871**
	4	24.997999	23.958036	**25.078997**	0.823245	0.788848	**0.824604**
	5	27.028038	25.604699	26.787410	0.871580	0.853626	0.869713
Starfish	2	**21.369702**	**21.369702**	**21.369702**	**0.614332**	**0.614332**	**0.614332**
	3	20.123501	**20.162296**	20.123501	**0.608668**	0.606647	**0.608668**
	4	22.500317	**22.809191**	22.255093	0.701103	**0.703400**	0.696909
	5	24.353985	**24.497387**	24.047765	**0.764141**	0.764097	0.760456
Wherry	2	**22.026562**	**22.026562**	**22.026562**	**0.702176**	**0.702176**	**0.702176**
	3	20.863813	**21.049248**	20.933124	0.723334	**0.724314**	0.723455
	4	21.572407	**21.849040**	21.572407	0.769623	**0.772163**	0.769623
	5	21.823918	24.051523	**24.240900**	0.791609	0.797219	**0.802442**
Zebra	2	**23.499759**	**23.499759**	**23.499759**	**0.712384**	**0.712384**	**0.712384**
	3	20.961240	20.838270	**21.369299**	0.711783	0.715772	**0.717959**
	4	23.121191	22.884141	**23.225227**	0.791075	**0.793775**	0.792903
	5	24.825848	**27.754191**	24.706135	0.832541	**0.836699**	0.833892

The values in bold face indicate the best result.

Table 5
Comparison of the best thresholds obtained from the methods based on Kapur's entropy criterion for test images.

Image	m	Kapur's entropy		Thresholds	
		PSO	ABC	PSO	ABC
Test 1	2	11.686333	11.687605	121, 154	122, 154
	3	14.525489	14.528310	122, 152, 192	122, 152, 189
	4	17.216593	17.225570	93, 122, 152, 186	90, 122, 152, 189
	5	19.482773	19.727792	100, 120, 155, 188, 211	90, 122, 152, 179, 212
Test 2	2	6.167160	6.167160	147, 200	147, 200
	3	7.713998	7.713998	96, 141, 197	93, 141, 198
	4	8.981947	9.072759	102, 147, 190, 230	92, 141, 178, 234
	5	10.071449	10.098961	74, 117, 145, 181, 232	72, 108, 146, 177, 230
Test 3	2	12.310165	12.310229	104, 165	104, 164
	3	15.477667	15.488384	99, 148, 213	96, 152, 215
	4	18.400698	18.404300	84, 129, 175, 219	81, 127, 170, 216
	5	21.010304	21.119833	74, 101, 137, 180, 228	72, 109, 147, 186, 223
Test 4	2	1.616401	1.616401	33, 46	33, 46
	3	1.616401	1.616401	18, 25, 73	18, 25, 73
	4	1.616401	1.616401	14, 31, 58, 60	14, 31, 58, 60

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 6
PSNR and SSIM metrics of the test images segmented with the thresholds yielding the best Kapur's entropy.

Image	m	PSNR (dB)		SSIM	
		PSO	ABC	PSO	ABC
Test image 1	2	20.997632	20.978213	0.872967	0.875996
	3	24.878019	24.853183	0.915911	0.912255
	4	28.289560	28.590540	0.915174	0.923294
	5	29.274019	30.335671	0.910701	0.921231
Test image 2	2	21.295226	21.295226	0.621754	0.621754
	3	25.750751	25.998110	0.754011	0.760274
	4	26.127076	27.510790	0.771649	0.808953
	5	29.534412	29.920916	0.845432	0.852922
Test image 3	2	23.389673	23.382350	0.729339	0.725902
	3	24.125128	24.114049	0.761142	0.767518
	4	26.372744	26.440209	0.860325	0.859555
	5	27.089076	27.686356	0.882444	0.896022
Test image 4	2	12.350280	12.350280	0.770204	0.770204
	3	12.350280	12.350280	0.770204	0.770204
	4	12.350280	12.350280	0.770204	0.770204

Table 7
Comparison of the best thresholds obtained from the methods based on Kapur's entropy criterion for real images.

Image	m	Kapur's entropy		Thresholds	
		PSO	ABC	PSO	ABC
Butterfly	2	12.649601	12.650776	114, 175	114, 174,
	3	15.742424	15.744942	115, 170, 218	114, 170, 219,
	4	18.751490	18.814762	79, 124, 171, 224	74, 118, 171, 219,
	5	21.347923	21.477840	69, 102, 131, 167, 220	72, 115, 147, 178, 220
Cameraman	2	12.168534	12.168753	129, 193	128, 193
	3	15.207067	15.227394	42, 107, 197	44, 104, 193
	4	18.365287	18.395542	44, 98, 140, 194	44, 97, 146, 197
	5	20.996574	21.144453	28, 84, 124, 160, 197	25, 61, 100, 146, 197
Cloud	2	12.679571	12.679798	89, 172	89, 170
	3	15.850579	15.851766	83, 144, 198,	82, 144, 198
	4	18.660191	18.692726	74, 125, 174, 216,	76, 126, 168, 210
	5	21.259996	21.368306	77, 109, 144, 172, 221	75, 108, 144, 180, 216
Elephants	2	12.128516	12.130062	119, 172	120, 171
	3	15.179607	15.192880	123, 163, 215	122, 164, 211
	4	17.928316	18.146352	84, 128, 152, 209	84, 124, 163, 211
	5	20.533222	20.794559	85, 121, 145, 179, 224	84, 124, 160, 192, 223
Landscape	2	11.098532	11.098550	98, 149	98, 150
	3	13.933832	14.013473	107, 166, 194	99, 165, 196
	4	16.574299	16.707206	97, 139, 172, 193	76, 123, 165, 196
	5	18.951146	19.227443	74, 101, 134, 176, 193	28, 59, 102, 165, 196
Lena	2	12.347015	12.347015	97, 164	97, 164
	3	15.316510	15.318053	83, 127, 178	82, 126, 175
	4	17.984092	18.012432	77, 115, 148, 180	64, 97, 138, 179
	5	20.456521	20.610566	72, 96, 122, 153, 190	63, 94, 128, 163, 194
Ostrich	2	12.611519	12.612476	120, 180	119, 180
	3	15.667314	15.672157	74, 124, 180	75, 123, 183
	4	18.562874	18.614032	73, 118, 172, 210	30, 77, 123, 183
	5	21.305444	21.534264	30, 79, 119, 171, 227	30, 75, 119, 159, 201
Plane	2	11.154947	11.154947	66, 101	66, 101
	3	14.010854	14.040908	35, 75, 113	35, 72, 102
	4	16.472249	16.762741	28, 70, 93, 160	35, 72, 102, 158
	5	19.067470	19.230072	33, 79, 102, 132, 158	34, 66, 96, 121, 158
Snake	2	12.434602	12.434921	84, 171	85, 171
	3	15.686447	15.695429	78, 146, 196	79, 143, 195
	4	18.703213	18.712623	76, 121, 165, 211	75, 124, 169, 213
	5	21.460950	21.504462	71, 112, 148, 184, 224	64, 103, 143, 180, 219
Starfish	2	12.999128	12.999386	89, 169	90, 169
	3	16.164480	16.166488	75, 130, 187	75, 130, 184
	4	19.109647	19.118751	71, 116, 161, 203	67, 115, 163, 206
	5	21.827982	21.877330	51, 89, 123, 168, 215	56, 95, 133, 172, 211
Wherry	2	12.204134	12.204231	108, 168	110, 169
	3	15.191377	15.194587	81, 121, 176	79, 121, 175
	4	18.037793	18.107350	55, 118, 151, 193	78, 118, 152, 193
	5	20.659751	20.825709	79, 113, 158, 190, 237	51, 85, 120, 152, 193
Zebra	2	12.272398	12.272881	98, 161	97, 161
	3	15.276590	15.280805	90, 139, 185	91, 137, 180
	4	18.140054	18.151984	88, 132, 171, 216	89, 132, 167, 209
	5	20.687239	20.751924	86, 122, 162, 197, 226	73, 103, 138, 172, 212

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 8
PSNR and SSIM metrics of the real images segmented with the thresholds yielding the best Kapur's entropy.

Image	m	PSNR (dB)		SSIM	
		PSO	ABC	PSO	ABC
Butterfly	2	21.539940	21.543635	0.696695	0.697747
	3	22.316838	22.354720	0.719402	0.718003
	4	26.165864	26.288471	0.807476	0.808660
	5	27.267429	27.213775	0.824348	0.825934
Cameraman	2	17.949998	18.066219	0.670583	0.673554
	3	22.633365	22.491244	0.824796	0.822744
	4	26.801473	26.935044	0.869142	0.850119
	5	28.632047	27.647027	0.865602	0.866324
Cloud	2	22.977042	23.019169	0.812871	0.811942
	3	25.022727	24.971105	0.845120	0.843608
	4	25.641133	25.993268	0.845352	0.855015
	5	26.926810	27.303576	0.863884	0.868857
Elephants	2	22.330109	22.284208	0.852787	0.855515
	3	25.321231	25.501893	0.842075	0.839765
	4	25.887471	26.385228	0.855042	0.849995
	5	27.778204	28.900542	0.859488	0.856065
Landscape	2	19.964328	20.027055	0.774236	0.774132
	3	20.432751	20.615742	0.807640	0.796096
	4	21.267059	23.316869	0.808160	0.805555
	5	24.088929	27.040758	0.831165	0.911279
Lena	2	22.816979	22.816979	0.685533	0.685533
	3	25.863818	25.959130	0.740139	0.740643
	4	28.132421	27.472782	0.780630	0.772646
	5	29.052412	28.784054	0.790301	0.787137
Ostrich	2	20.584207	20.616952	0.774941	0.775542
	3	26.708803	26.681147	0.867422	0.865942
	4	27.086271	26.949691	0.873118	0.868793
	5	27.055509	27.625401	0.869110	0.881062
Plane	2	26.999238	26.999238	0.922988	0.922988
	3	28.852117	27.465706	0.919635	0.926414
	4	26.500038	27.505245	0.936616	0.926533
	5	30.738801	30.885264	0.901590	0.913625
Snake	2	22.085744	22.148286	0.638025	0.642460
	3	23.623973	23.940914	0.713947	0.729560
	4	26.142313	25.821976	0.815056	0.803648
	5	27.290477	27.395175	0.846936	0.847403
Starfish	2	21.246847	21.245456	0.607571	0.607906
	3	24.019582	24.039595	0.692250	0.693578
	4	25.538598	25.535651	0.741076	0.737633
	5	27.224511	27.270102	0.787343	0.785400
Wherry	2	21.380551	21.405756	0.722998	0.722530
	3	22.555976	22.487742	0.765967	0.765833
	4	24.414930	25.415332	0.773443	0.800391
	5	28.341669	25.764466	0.815860	0.818221
Zebra	2	23.384534	23.385387	0.716809	0.715501
	3	25.745022	25.749305	0.761082	0.768848
	4	26.909854	26.945864	0.781371	0.786364
	5	27.787581	28.865083	0.804721	0.827451

Table 9
A statistical analysis on PSNR and SSIM values obtained from the images segmented by the methods. BCV: between-class variance, Std.Dev.: standard deviation, p: probability of the result, $h = 0$ indicates null hypothesis cannot be rejected at the 0.05 significance level.

	Alg	BCV			Entropy			BCV-entropy
		Mean	*Std.Dev.*	h/p(ABC − Alg)	Mean	*Std.Dev.*	h/p(ABC − Alg)	h/p
PSNR	Otsu	24.090861	2.439522	0/0.7723	–	–	–	–
	PSO	24.118776	2.441506	0/0.8154	24.9797401	2.8337906	0/0.8269	0/0.1140
	ABC	24.235517	2.486021	–	25.1057756	2.761951		0/0.1082
SSIM	Otsu	0.8027887	0.0781	0/0.9665	–	–	–	–
	PSO	0.8004121	0.076753	0/0.8467	0.79838213	0.0738489	0/0.8622	0/0.8954
	ABC	0.8034544	0.077828	–	0.80105196	0.0749414	–	0/0.8782

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

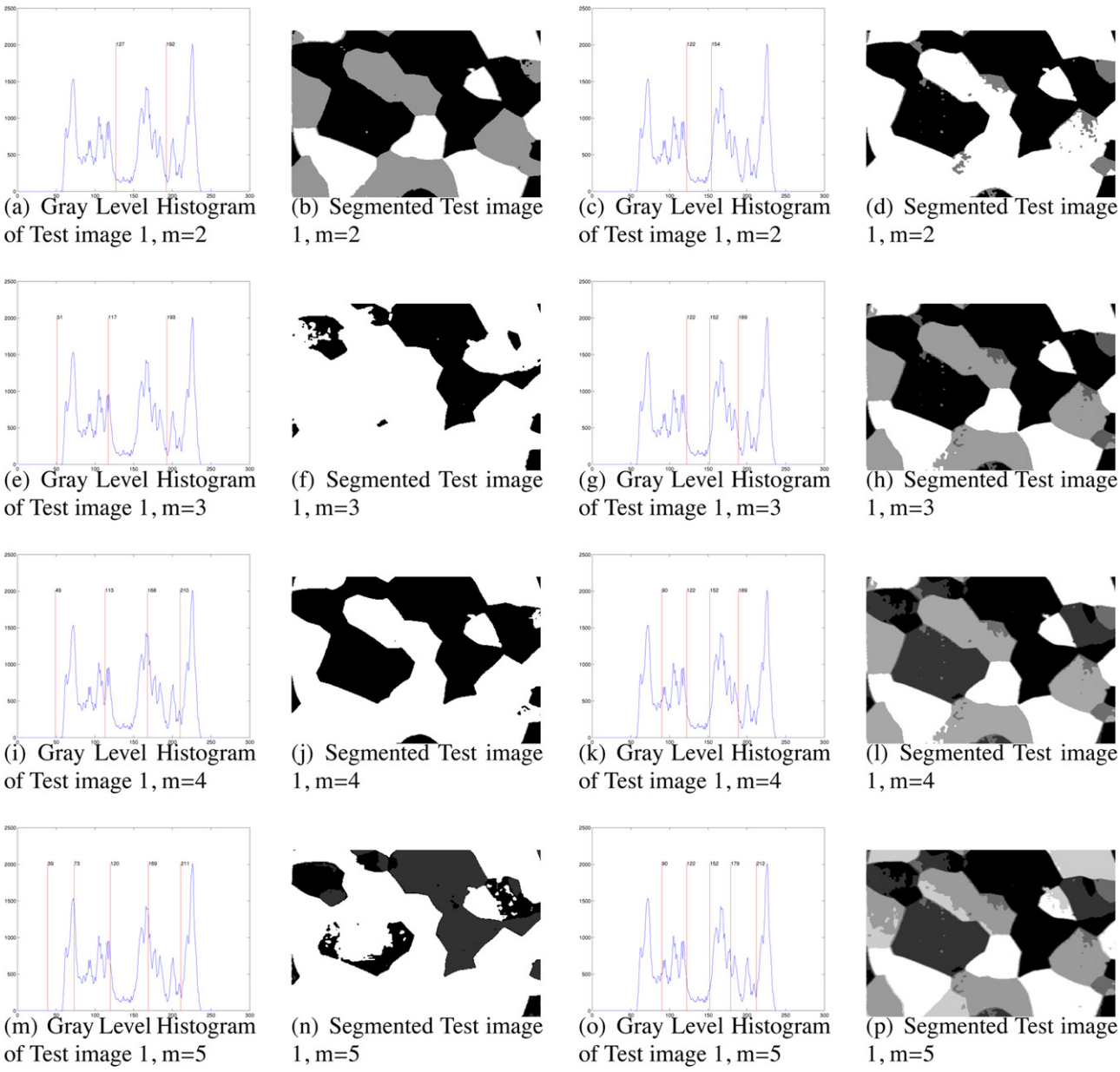


Fig. 3. Test image 1 for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of test image 1, $m = 2$, (b) segmented test image 1, $m = 2$, (c) gray level histogram of test image 1, $m = 2$, (d) segmented test image 1, $m = 2$, (e) gray level histogram of test image 1, $m = 3$, (f) segmented test image 1, $m = 3$, (g) gray level histogram of test image 1, $m = 3$, (h) segmented test image 1, $m = 3$, (i) gray level histogram of test image 1, $m = 4$, (j) segmented test image 1, $m = 4$, (k) gray level histogram of test image 1, $m = 4$, (l) segmented test image 1, $m = 4$, (m) gray level histogram of test image 1, $m = 5$, (n) segmented test image 1, $m = 5$, (o) gray level histogram of test image 1, $m = 5$, and (p) segmented test image 1, $m = 5$.

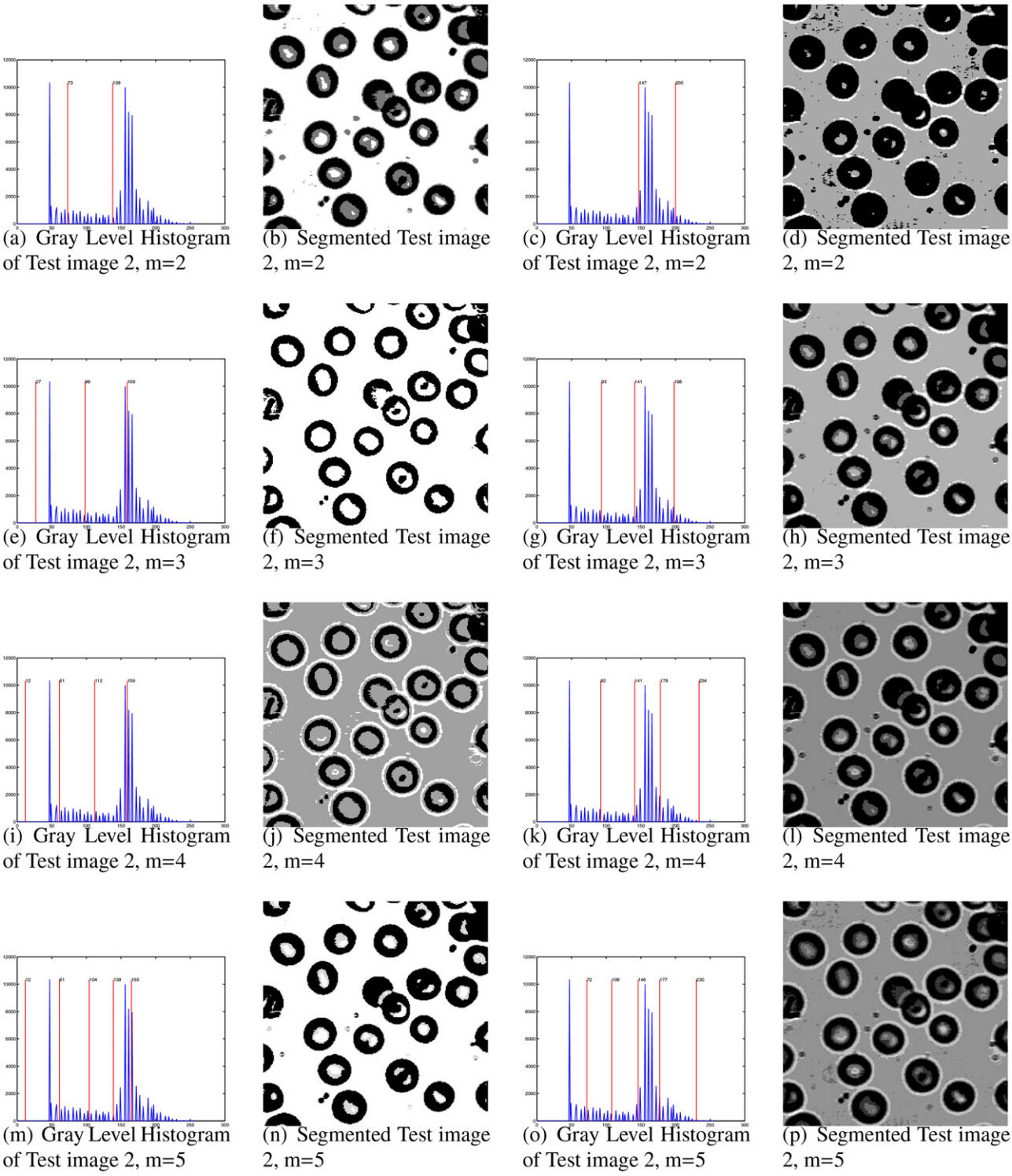
The same experiments were repeated for the images in Fig. 2. In Table 3, the best solutions of the algorithms for 30 runs and the between-class variance values associated with the best solutions have been presented. When the number of thresholds is two, all of the algorithms deliver equal performance. For all other cases, the ABC algorithm produces the minimum between-class variance value. For the Lena image ($m = 2$–4), results obtained from the ABC algorithm and Otsu's method are equal. The segmented images are constructed by assigning each class a grey level value calculated from the mean of the members in the class. PSNR and SSIM metrics of the segmented images, given in Fig. 2, have been recorded in Table 4. There seems to be a low correlation between PSNR and the between-class variance.

Segmented images and grey level histograms labelled with thresholds have been given in Figs. 3–13 for some images. The first columns of the images show the histograms labelled with the thresholds found by the ABC algorithm on the between-class variance, and the second columns give the segmented images through the threshold values obtained from the ABC algorithm on the between-class variance fitness function.

4.3. Experiment 2: Maximising entropy

In the second part of the experiments, Kapur's entropy has been given to the algorithms as the fitness function to be maximised. The best multilevel thresholds obtained from the algorithms after

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072



(a) Gray Level Histogram of Test image 2, m=2 (b) Segmented Test image 2, m=2 (c) Gray Level Histogram of Test image 2, m=2 (d) Segmented Test image 2, m=2

(e) Gray Level Histogram of Test image 2, m=3 (f) Segmented Test image 2, m=3 (g) Gray Level Histogram of Test image 2, m=3 (h) Segmented Test image 2, m=3

(i) Gray Level Histogram of Test image 2, m=4 (j) Segmented Test image 2, m=4 (k) Gray Level Histogram of Test image 2, m=4 (l) Segmented Test image 2, m=4

(m) Gray Level Histogram of Test image 2, m=5 (n) Segmented Test image 2, m=5 (o) Gray Level Histogram of Test image 2, m=5 (p) Segmented Test image 2, m=5

Fig. 4. Test image 2 for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of test image 2, $m = 2$, (b) segmented test image 2, $m = 2$, (c) gray level histogram of test image 2, $m = 2$, (d) segmented test image 2, $m = 2$, (e) gray level histogram of test image 2, $m = 3$, (f) segmented test image 2, $m = 3$, (g) gray level histogram of test image 2, $m = 3$, (h) segmented test image 2, $m = 3$, (i) gray level histogram of test image 2, $m = 4$, (j) segmented test image 2, $m = 4$, (k) gray level histogram of test image 2, $m = 4$, (l) segmented test image 2, $m = 4$, (m) gray level histogram of test image 2, $m = 5$, (n) segmented test image 2, $m = 5$, (o) gray level histogram of test image 2, $m = 5$, and (p) segmented test image 2, $m = 5$.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

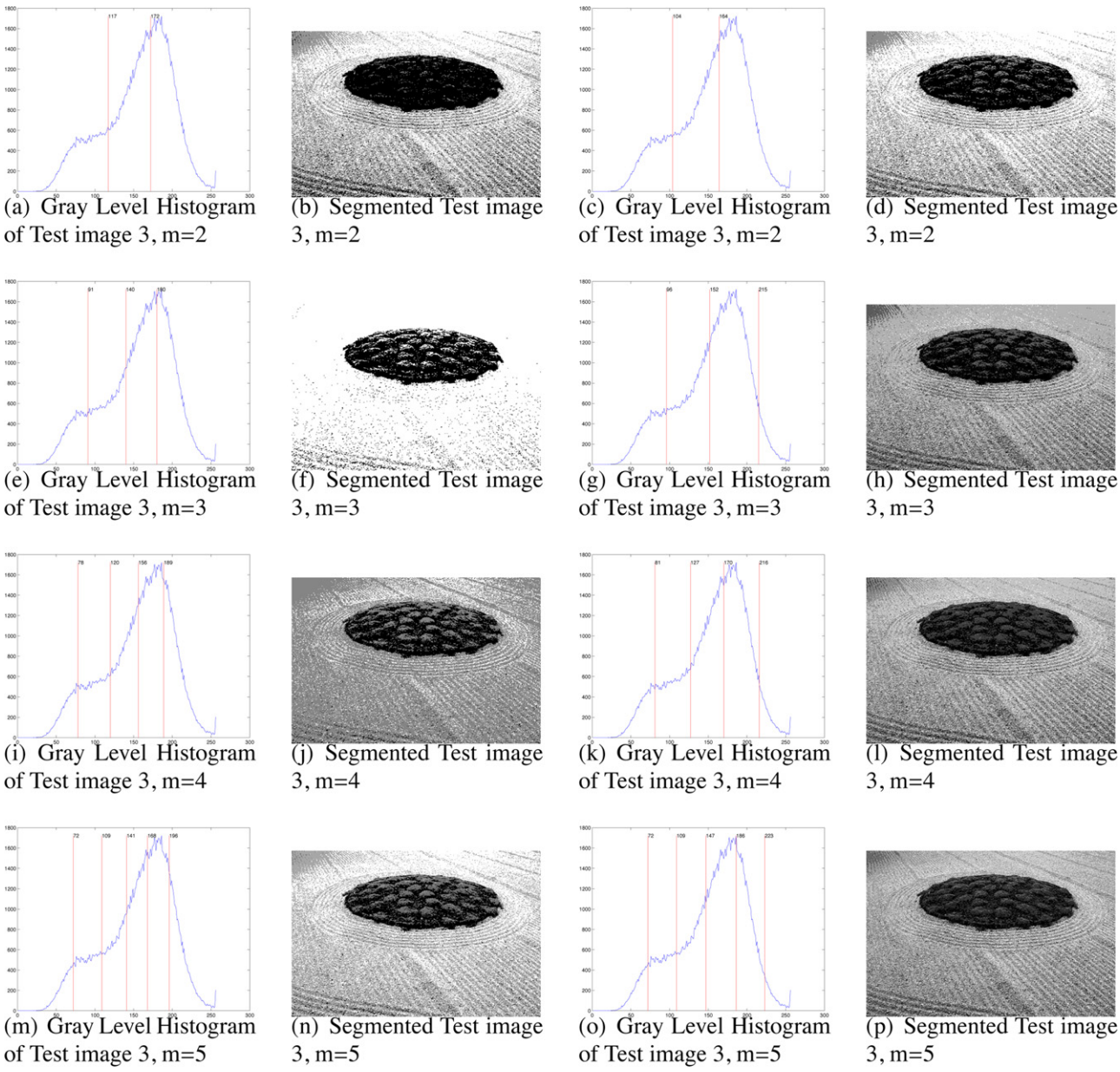


Fig. 5. Test image 3 for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of test image 3, $m = 2$, (b) segmented test image 3, $m = 2$, (c) gray level histogram of test image 3, $m = 2$, (d) segmented test image 3, $m = 2$, (e) gray level histogram of test image 3, $m = 3$, (f) segmented test image 3, $m = 3$, (g) gray level histogram of test image 3, $m = 3$, (h) segmented test image 3, $m = 3$, (i) gray level histogram of test image 3, $m = 4$, (j) segmented test image 3, $m = 4$, (k) gray level histogram of test image 3, $m = 4$, (l) segmented test image 3, $m = 4$, (m) gray level histogram of test image 3, $m = 5$, (n) segmented test image 3, $m = 5$, (o) gray level histogram of test image 3, $m = 5$, and (p) segmented test image 3, $m = 5$.

30 runs and the corresponding Kapur's entropy values are reported in Table 5 for simple images and in Table 7 for the images in Fig. 2.

PSO and ABC algorithms that rely on the entropy-based fitness function cannot find satisfactory results for test image 4 in Fig. 1 in terms of entropy. Furthermore, they have very low PSNR and SSIM values in Table 6. It can be seen from Fig. 6 that the histogram of this image has only five bars. Due to the nature of Kapur's entropy, it provides a centralised distribution and is maximal when the members of the classes are equally probable, it may be difficult for algorithms based on entropy to find the thresholds from the image histograms with only bars. To overcome this situation, algorithms can be tailored to check the number of bars in the histogram.

Table 7.

The ABC algorithm produces higher Kapur's entropy on all images in Fig. 2 except for the Lena image with $m = 2$ and the Plane image with $m = 2$ on which the PSO algorithm yields the same entropy with the ABC algorithm.

The third columns of Figs. 3–13 present the histograms labelled with the thresholds produced by the ABC algorithm on the entropy criterion, and the forth columns give the images segmented by the thresholds obtained from the ABC algorithm on the entropy criterion. The histograms in the third columns of Figs. 7–13 show that the optimal threshold values produced by the ABC algorithm tend to split a histogram considering the separability and compactness of the groups, which arise from using Kapur's entropy as the objective function.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

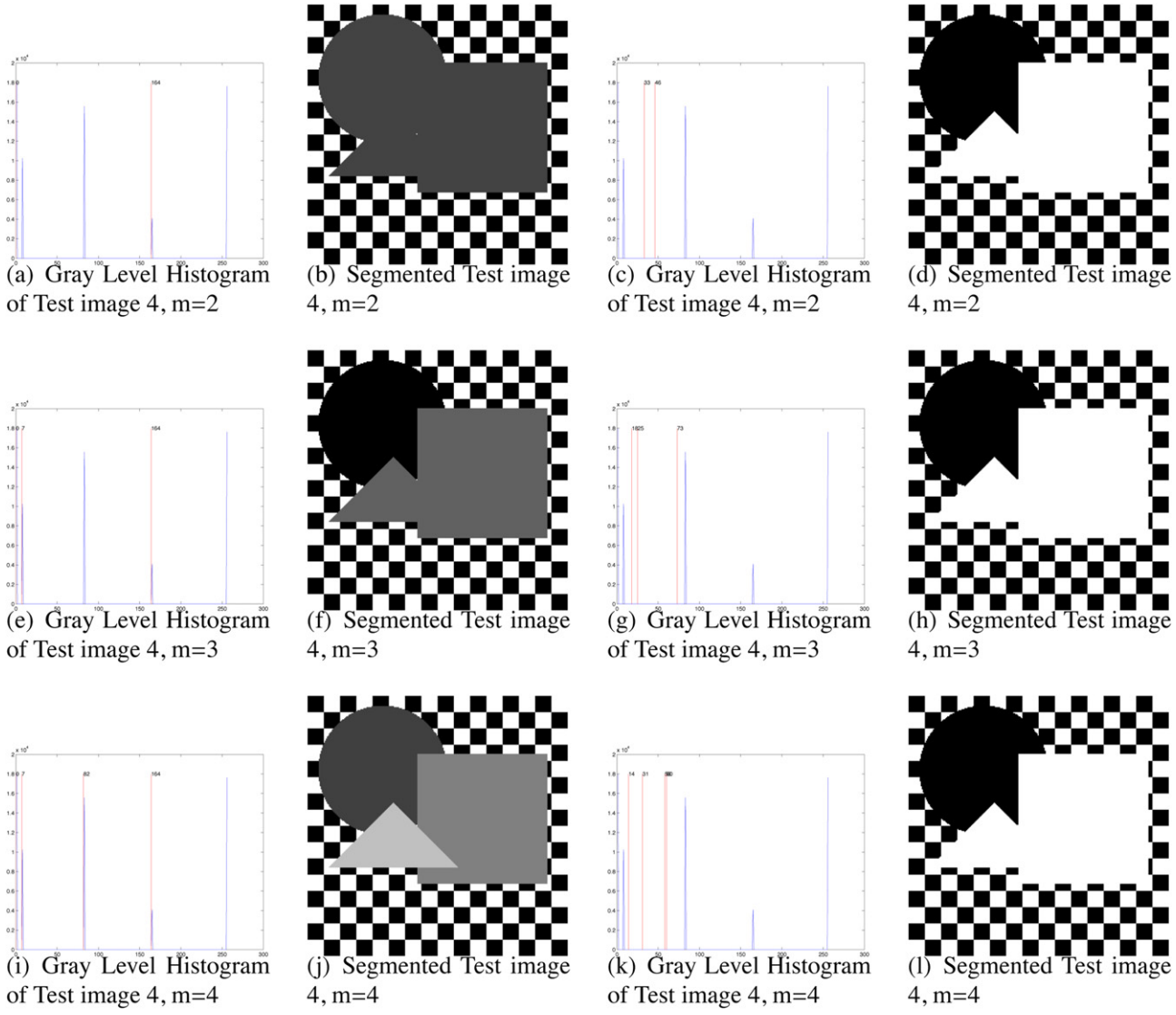


Fig. 6. Test image 4 for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of test image 4, $m = 2$, (b) segmented test image 4, $m = 2$, (c) gray level histogram of test image 4, $m = 2$, (d) segmented test image 4, $m = 2$, (e) gray level histogram of test image 4, $m = 3$, (f) segmented test image 4, $m = 3$, (g) gray level histogram of test image 4, $m = 3$, (h) segmented test image 4, $m = 3$, (i) gray level histogram of test image 4, $m = 4$, (j) segmented test image 4, $m = 4$, (k) gray level histogram of test image 4, $m = 4$, and (l) segmented test image 4, $m = 4$.

PSNR and SSIM values of the best results are reported in Table 8. Although the PSO algorithm does not yield the best Kapur's entropy, it yields the best PSNR in some cases. They may not always be parallel according to the histogram of the original image because the entropy considers the distribution of grey levels and preserves the information content on this basis while PSNR considers only the mean square error of grey levels. Table 8 also shows that when the number of thresholds approximates the number of different grey levels in the image, SSIM is very high (Landscape, m=5). When the number of thresholds exceeds the number of grey levels in the image, SSIM again gets worse (Plane, $m = 3$–5).

4.4. A discussion on the metrics

To evaluate if there is a significant difference between the PSNR and SSIM values of the best solutions for all images and m values, statistical results have been presented in Table 9. In the table, mean and standard deviation values of PSNR and SSIM metrics for Otsu's method on the between-class variance and for PSO and ABC algorithms on both the between-class variance and Kapur's entropy have been reported. A paired t-test has been used in the analysis because PSNR and SSIM values of all pairs of the algorithms are normally distributed. The ABC algorithm has been chosen as the control algorithm. In the table, p is the probability of the test result, and h indicates the result of the test. A zero value of the h reports that the means are equal at the 0.05 significance level; otherwise, the null hypothesis can be rejected at the 0.05 level. Statistical results in Table 9 suggest that PSNR values of the best solutions coming from the ABC algorithm and the Otsu method, based on the between-class variance, have equal means, as well as those from the ABC and PSO algorithms based on the between-class variance. SSIM values also have equal means, with a probability of 0.9665 for the ABC–Otsu pair and with a probability of 0.8467 for the ABC–PSO pair. Similarly, the ABC and PSO pair on Kapur's entropy has equal means in terms of PSNR and SSIM.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

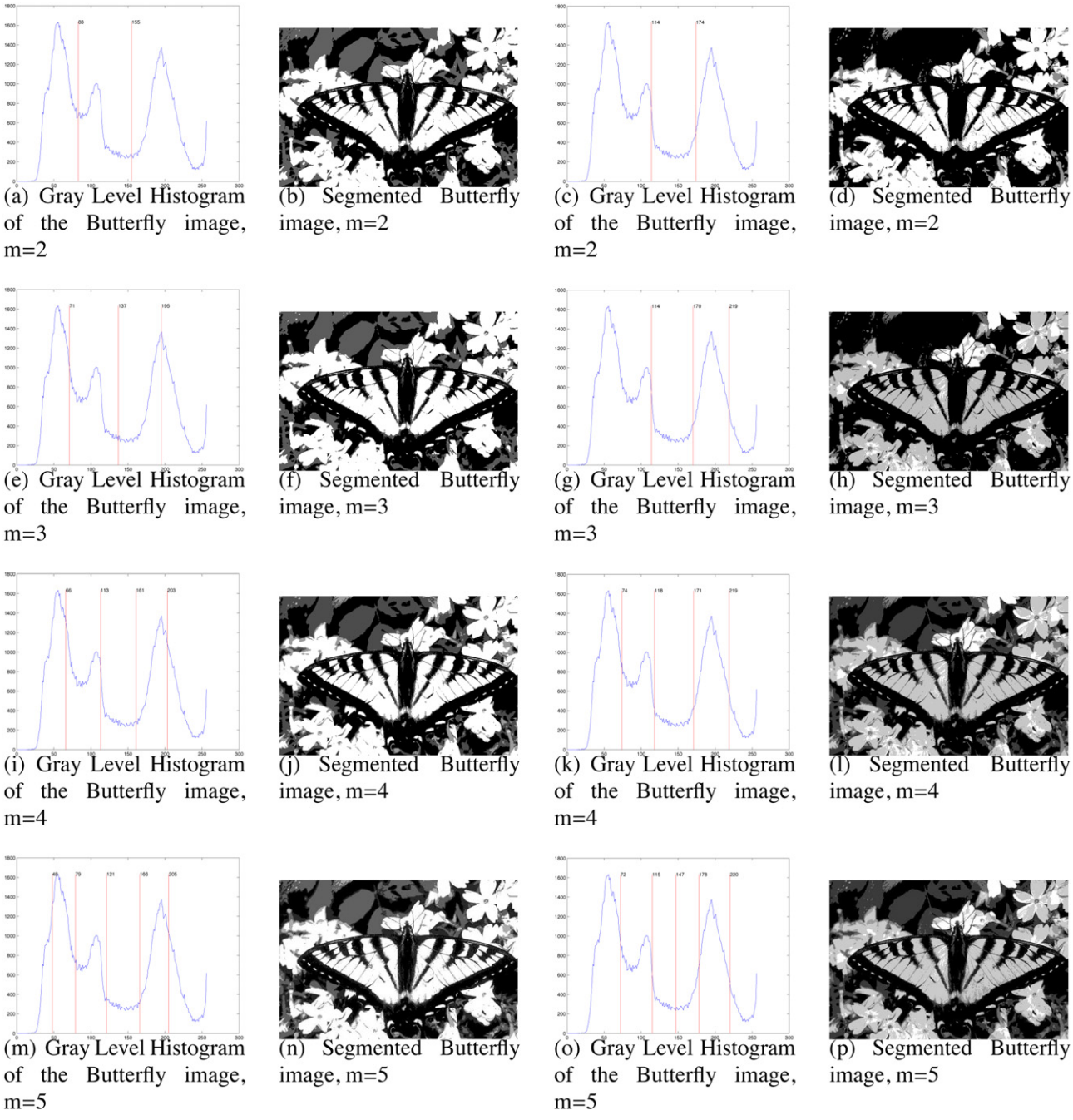


Fig. 7. The Butterfly image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (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, $m = 5$, and (p) segmented Butterfly image, $m = 5$.

Another test checks if there is a significant difference between the fitness functions employed in terms of PSNR and SSIM values. PSNR values of the ABC algorithm based on the between-class variance and PSNR values of the ABC algorithm based on Kapur's entropy have been compared to see if they come from distributions with equal means. The same test is also repeated for the PSO algorithm, and the results have been reported on the last column of Table 9. The test indicates that PSNR and SSIM values coming from ABC algorithms with different fitness functions have equal means, with probabilities of 0.1082 and 0.8782, respectively. The same test for PSO provides a similar result.

In addition to the tests that check the differences between the algorithms, the relations among PSNR, SSIM, entropy and the between-class variance have also been investigated. To evaluate this relation quantitatively, in Table 10, correlation coefficients between metrics and fitness functions have been presented. The table shows a low correlation between PSNR and the between-class variance. The correlation coefficient between PSNR and entropy is

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

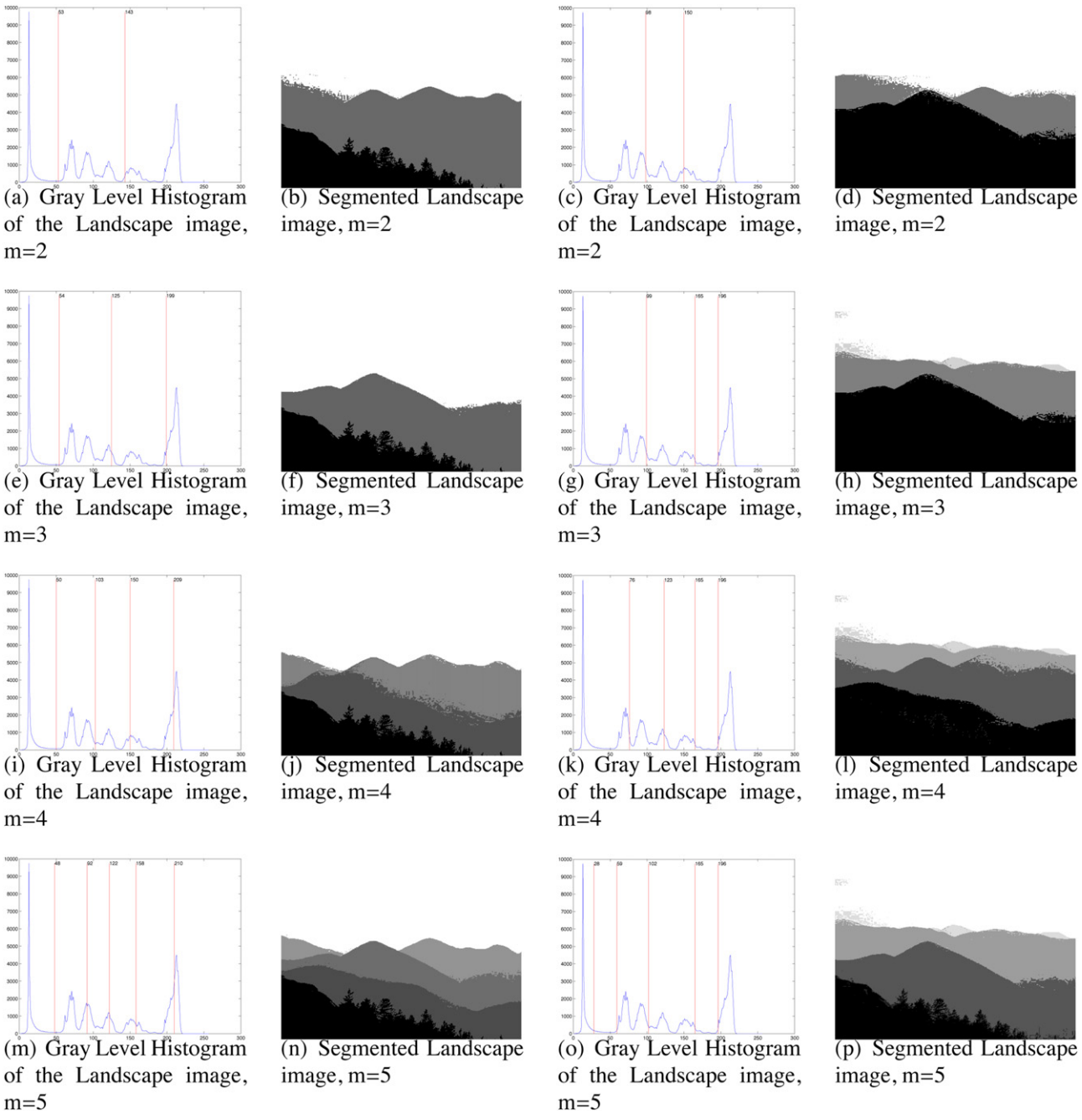


Fig. 8. The Landscape image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (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, $m = 5$, and (p) segmented Landscape image, $m = 5$.

Table 10
Correlation coefficients between the methods and the metrics.

	PSNR	SSIM
BCV	0.2114	0.4102
Entropy	0.7472	0.4217

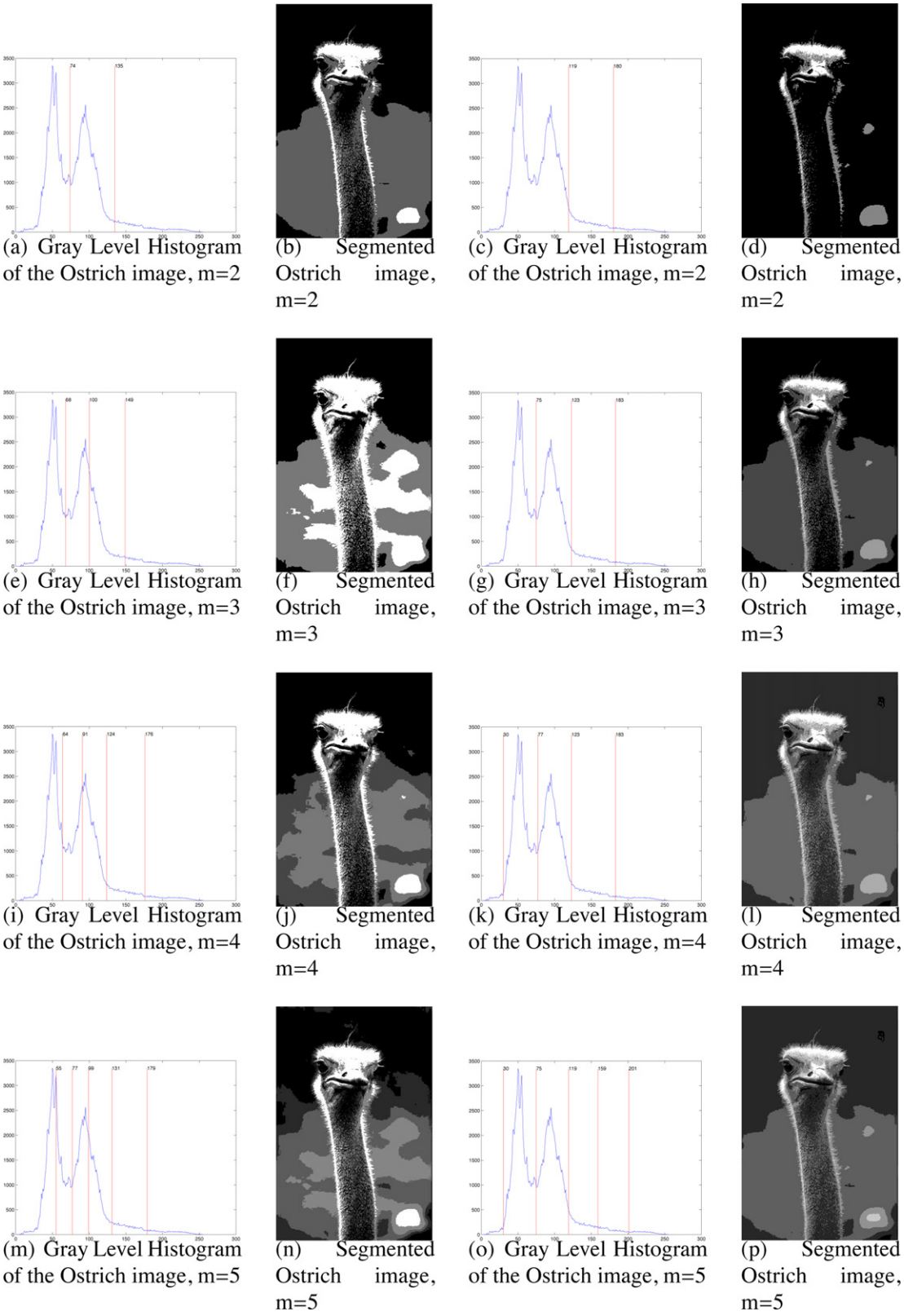
higher than that between PSNR and the between-class variance. The correlation between SSIM and the entropy, as well as the correlation between SSIM and the between-class variance, can be said to be close to the average.

High PSNR values may be obtained from methods that minimise MSE by assigning each pixel to a group with the least distance to the pixel value. PSNR values may not be maximal through the methods considering histogram distributions, such as entropy and the between-class variance.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Fig. 9. The Lena image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (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, $m = 5$, and (p) segmented Lena image, $m = 5$.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072



(a) Gray Level Histogram of the Ostrich image, m=2 (b) Segmented Ostrich image, m=2 (c) Gray Level Histogram of the Ostrich image, m=2 (d) Segmented Ostrich image, m=2

(e) Gray Level Histogram of the Ostrich image, m=3 (f) Segmented Ostrich image, m=3 (g) Gray Level Histogram of the Ostrich image, m=3 (h) Segmented Ostrich image, m=3

(i) Gray Level Histogram of the Ostrich image, m=4 (j) Segmented Ostrich image, m=4 (k) Gray Level Histogram of the Ostrich image, m=4 (l) Segmented Ostrich image, m=4

(m) Gray Level Histogram of the Ostrich image, m=5 (n) Segmented Ostrich image, m=5 (o) Gray Level Histogram of the Ostrich image, m=5 (p) Segmented Ostrich image, m=5

Fig. 10. The Ostrich image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of the Ostrich image, $m = 2$, (b) segmented Ostrich image, $m = 2$, (c) gray level histogram of the Ostrich image, $m = 2$, (d) segmented Ostrich image, $m = 2$, (e) gray level histogram of the Ostrich image, $m = 3$, (f) segmented Ostrich image, $m = 3$, (g) gray level histogram of the Ostrich image, $m = 3$, (h) segmented Ostrich image, $m = 3$, (i) gray level histogram of the Ostrich image, $m = 4$, (j) segmented Ostrich image, $m = 4$, (k) gray level histogram of the Ostrich image, $m = 4$, (l) segmented Ostrich image, $m = 4$, (m) gray level histogram of the Ostrich image, $m = 5$, (n) segmented Ostrich image, $m = 5$, (o) gray level histogram of the Ostrich image, $m = 5$, and (p) segmented Ostrich image, $m = 5$.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

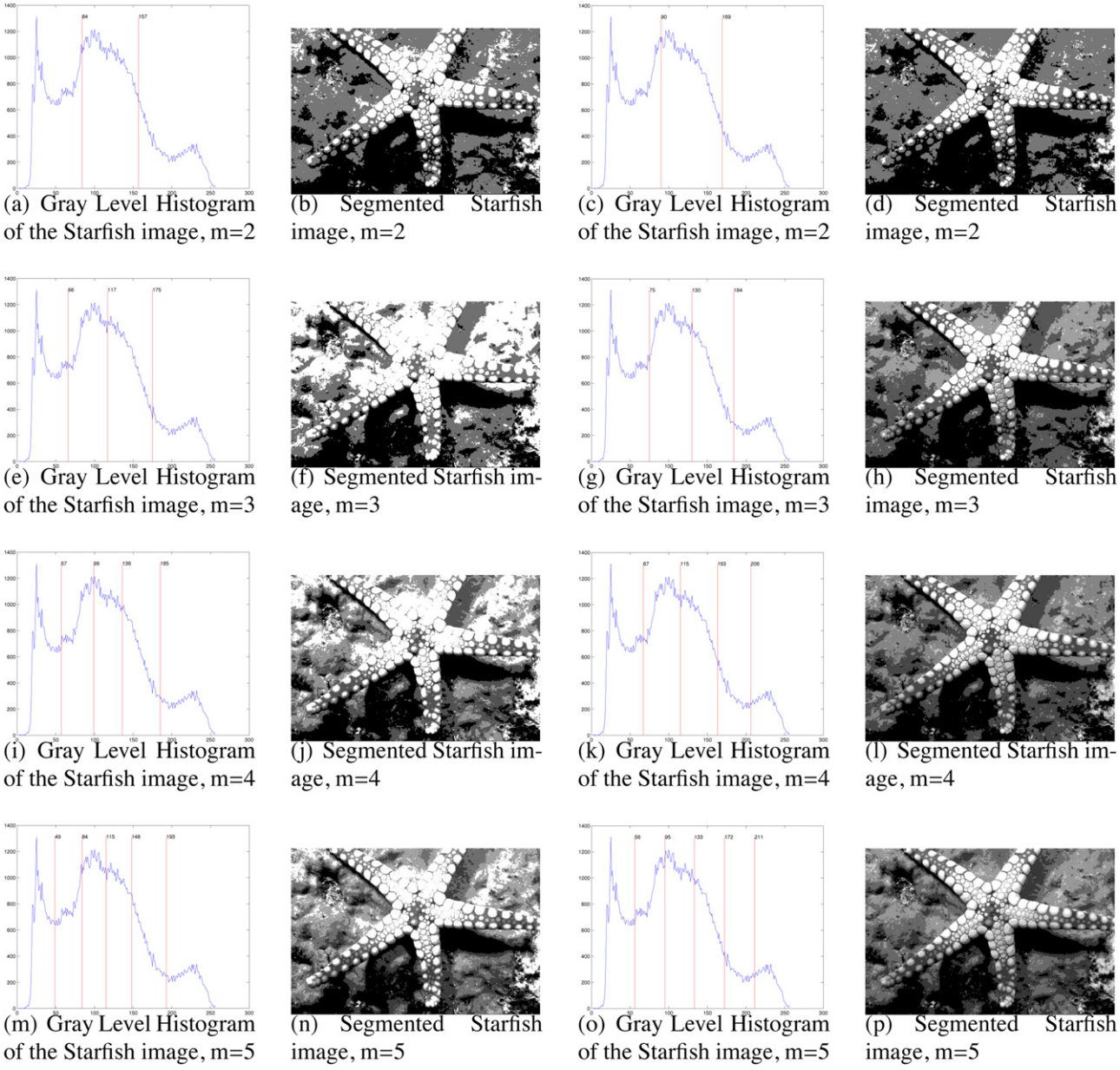


Fig. 11. The Starfish image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (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, $m = 5$, and (p) segmented Starfish image, $m = 5$.

A fitness function with more objectives can be used in the optimisation. There are some multi-objective approaches that are based on a pareto-approach that require prior knowledge and a designer decision to select one from the alternatives in pareto space. This may not be practical in automatic unsupervised thresholding. Another type of multi-objective optimisation is penalty-based algorithms, which transform the multiple objectives into a single objective by weighting each objective. However, determining the optimal weighting factors is another optimisation problem. Moreover, this may increase the dimension of the thresholding problem when the thresholds and the weights are addressed simultaneously.

4.5. Statistical analysis

To analyse the efficiency and the stability of the algorithms, the mean and standard deviations of 30 runs have been presented in Table 11 for the experiments on the between-class variance and in Table 12 for the experiments on the Kapur's entropy. The number of cases investigated is 48 and $48 \times 30 = 1440$ runs have been analysed in total. The comparisons have been realised using statistical tools for hypothesis testing. The between-class variance and the entropy values coming from 30 runs of the algorithms failed normality tests; therefore, a nonparametric Wilcoxon rank sum test was applied to test if the algorithms have equal median values. In the experiments

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

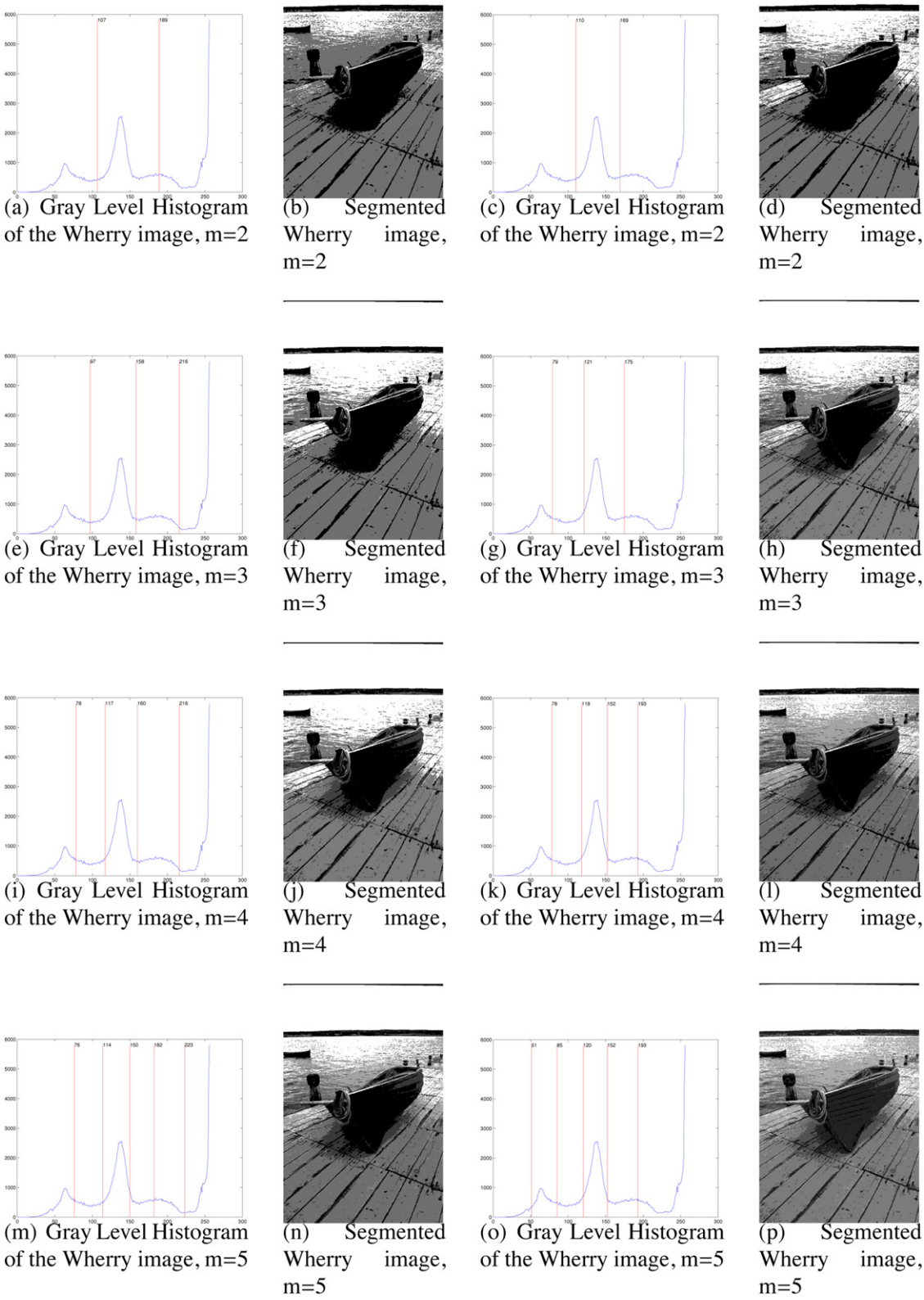


Fig. 12. The Wherry image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of the Wherry image, $m = 2$, (b) segmented Wherry image, $m = 2$, (c) gray level histogram of the Wherry image, $m = 2$, (d) segmented Wherry image, $m = 2$, (e) gray level histogram of the Wherry image, $m = 3$, (f) segmented Wherry image, $m = 3$, (g) gray level histogram of the Wherry image, $m = 3$, (h) segmented Wherry image, $m = 3$, (i) gray level histogram of the Wherry image, $m = 4$, (j) segmented Wherry image, $m = 4$, (k) gray level histogram of the Wherry image, $m = 4$, (l) segmented Wherry image, $m = 4$, (m) gray level histogram of the Wherry image, $m = 5$, (n) segmented Wherry image, $m = 5$, (o) gray level histogram of the Wherry image, $m = 5$, and (p) segmented Wherry image, $m = 5$.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

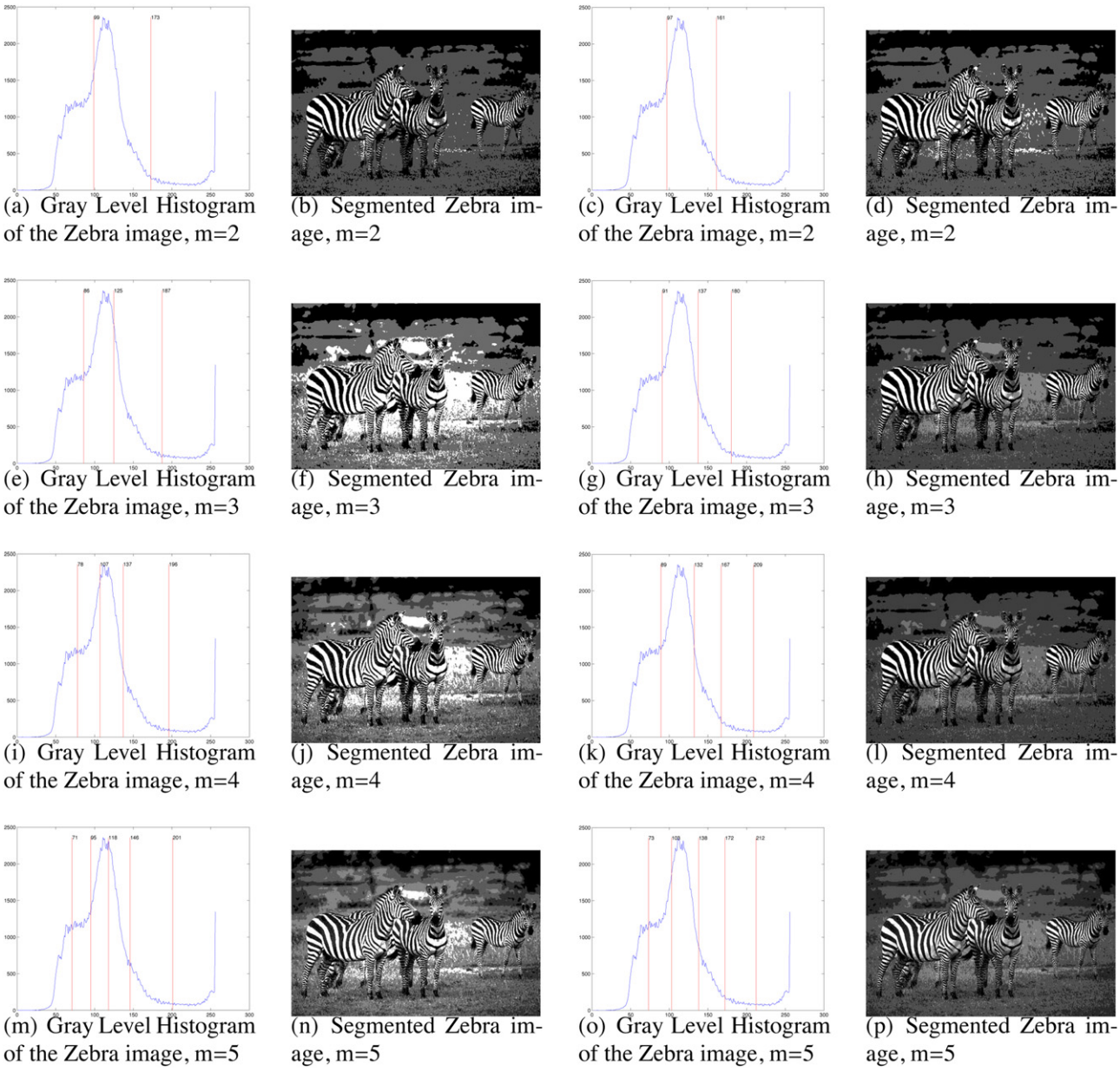


Fig. 13. The Zebra image for $m = 2$–5. First column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on the between-class variance criterion. Second column is the image segmented with the best thresholds obtained from the ABC algorithm based on the between-class variance criterion. Third column is the histogram of the image labeled with the best threshold values obtained from the ABC algorithm based on Kapur's entropy criterion. Forth column is the image segmented with the best thresholds obtained from the ABC algorithm based on Kapur's entropy criterion. (a) Gray level histogram of the Zebra image, $m = 2$, (b) segmented Zebra image, $m = 2$, (c) gray level histogram of the Zebra image, $m = 2$, (d) segmented Zebra image, $m = 2$, (e) gray level histogram of the Zebra image, $m = 3$, (f) segmented Zebra image, $m = 3$, (g) gray level histogram of the Zebra image, $m = 3$, (h) segmented Zebra image, $m = 3$, (i) gray level histogram of the Zebra image, $m = 4$, (j) segmented Zebra image, $m = 4$, (k) gray level histogram of the Zebra image, $m = 4$, (l) segmented Zebra image, $m = 4$, (m) gray level histogram of the Zebra image, $m = 5$, (n) segmented Zebra image, $m = 5$, (o) gray level histogram of the Zebra image, $m = 5$, and (p) segmented Zebra image, $m = 5$.

with between-class variance, the ABC algorithm was chosen as the control algorithm and was compared with the runner-up algorithm in terms of the mean value.

Table 11 states that the algorithms come from distributions with equal medians provided that the number of thresholds is two. The Lena image segmented with $m = 3$, the Snake image with $m = 4$ and the Zebra image with $m = 4$ are also cases in which the distributions have equal medians. The ABC algorithm produced better results in 33 cases out of 48 total cases.

In the experiments using the entropy criterion, the best solutions of the PSO algorithm were the same as those of the ABC algorithm on Lena and Plane images for $m = 2$. However, the means and the standard deviations for these cases in Table 12 are still worse than those for the ABC algorithm. The standard deviation of the PSO algorithm increases much more as the number of thresholds is increased. From the table, it is seen that the standard deviation of the ABC algorithm is very low for all cases, and the ABC algorithm is stable, which is a desired characteristic in real-time applications. Another advantage of the ABC algorithm is that it employs fewer control parameters than the PSO algorithm. The Wilcoxon rank sum test states that there is a significant difference between the median values of the PSO algorithm and the ABC algorithm; the ABC algorithm is better than the PSO algorithm.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 11
Statistical anaylsis of 30 runs for each of 48 cases for the experiments on the between-class variance criterion. p: probability of the statistic, $h = 0$ indicates null hypothesis cannot be rejected at the 0.05 significance level, $h = 1$ indicates null hypothesis can be rejected.

Image	m	Mean ± Std.Dev.			Wilcoxon rank sum	
		Otsu	PSO	ABC	p	h
Butterfly	2	0.923426 ± 0.000000	0.923426 ± 0.000000	0.923426 ± 0.000000	NaN	0
	3	0.951457 ± 0.000000	0.941194 ± 0.005211	0.951471 ± 0.000000	0.000	1
	4	0.964544 ± 0.000000	0.956956 ± 0.004427	0.965881 ± 0.000018	0.000	1
	5	0.975649 ± 0.000000	0.966415 ± 0.002906	0.975765 ± 0.000051	0.000	1
Cameraman	2	0.939251 ± 0.000000	0.939251 ± 0.000000	0.939251 ± 0.000000	NaN	0
	3	0.958523 ± 0.000000	0.951411 ± 0.004760	0.958641 ± 0.000011	0.000	1
	4	0.970693 ± 0.000000	0.963493 ± 0.004040	0.972781 ± 0.000013	0.000	1
	5	0.978398 ± 0.000000	0.971944 ± 0.003647	0.980814 ± 0.000038	0.000	1
Cloud	2	0.910900 ± 0.000000	0.910900 ± 0.000000	0.910900 ± 0.000000	NaN	0
	3	0.945091 ± 0.000000	0.934510 ± 0.007980	0.945211 ± 0.000000	0.000	1
	4	0.956891 ± 0.000000	0.950617 ± 0.005045	0.962218 ± 0.000016	0.000	1
	5	0.974485 ± 0.000000	0.961779 ± 0.005133	0.974493 ± 0.000039	0.000	1
Elephants	2	0.914539 ± 0.000000	0.914539 ± 0.000000	0.914539 ± 0.000000	NaN	0
	3	0.934884 ± 0.000000	0.934640 ± 0.006151	0.944901 ± 0.000014	0.000	1
	4	0.964154 ± 0.000000	0.952068 ± 0.004994	0.964218 ± 0.000008	0.000	1
	5	0.969813 ± 0.000000	0.960751 ± 0.004259	0.974052 ± 0.000036	0.000	1
Landscape	2	0.913001 ± 0.000000	0.913001 ± 0.000000	0.913001 ± 0.000000	NaN	0
	3	0.970362 ± 0.000000	0.962619 ± 0.006844	0.970678 ± 0.000000	0.000	1
	4	0.984290 ± 0.000000	0.976815 ± 0.004478	0.984402 ± 0.000000	0.000	1
	5	0.992186 ± 0.000000	0.983829 ± 0.003277	0.992228 ± 0.000002	0.000	1
Lena	2	0.856599 ± 0.000000	0.856599 ± 0.000000	0.856599 ± 0.000000	NaN	0
	3	0.929388 ± 0.000000	0.915878 ± 0.011051	0.929388 ± 0.000000	NaN	0
	4	0.957044 ± 0.000000	0.938762 ± 0.009880	0.957042 ± 0.000009	0.001	1
	5	0.968245 ± 0.000000	0.953234 ± 0.007690	0.968340 ± 0.000043	0.000	1
Ostrich	2	0.848439 ± 0.000000	0.848439 ± 0.000000	0.848439 ± 0.000000	NaN	0
	3	0.900417 ± 0.000000	0.872233 ± 0.016433	0.900453 ± 0.000000	0.000	1
	4	0.931263 ± 0.000000	0.905247 ± 0.011939	0.931564 ± 0.000026	0.000	1
	5	0.949724 ± 0.000000	0.923879 ± 0.011981	0.951366 ± 0.000107	0.000	1
Plane	2	0.856388 ± 0.000000	0.856388 ± 0.000000	0.856388 ± 0.000000	NaN	0
	3	0.919361 ± 0.000000	0.898420 ± 0.017213	0.920436 ± 0.000000	0.000	1
	4	0.941907 ± 0.000000	0.921727 ± 0.008324	0.949268 ± 0.000033	0.000	1
	5	0.957237 ± 0.000000	0.939891 ± 0.007346	0.963647 ± 0.000073	0.000	1
Snake	2	0.803111 ± 0.000000	0.803111 ± 0.000000	0.803111 ± 0.000000	NaN	0
	3	0.883920 ± 0.000000	0.851309 ± 0.022163	0.884028 ± 0.000000	0.000	1
	4	0.923659 ± 0.000000	0.886316 ± 0.012692	0.923644 ± 0.000052	0.814	0
	5	0.945009 ± 0.000000	0.915173 ± 0.012079	0.945442 ± 0.000066	0.000	1
Starfish	2	0.843043 ± 0.000000	0.843043 ± 0.000000	0.843043 ± 0.000000	NaN	0
	3	0.920170 ± 0.000000	0.900588 ± 0.011555	0.920181 ± 0.000000	0.000	1
	4	0.948285 ± 0.000000	0.929377 ± 0.011356	0.948555 ± 0.000023	0.000	1
	5	0.963940 ± 0.000000	0.947685 ± 0.008680	0.964116 ± 0.000049	0.000	1
Wherry	2	0.890490 ± 0.000000	0.890490 ± 0.000000	0.890490 ± 0.000000	NaN	0
	3	0.952274 ± 0.000000	0.936779 ± 0.009660	0.952338 ± 0.000000	0.000	1
	4	0.967525 ± 0.000000	0.954071 ± 0.006611	0.967556 ± 0.000016	0.000	1
	5	0.972141 ± 0.000000	0.963420 ± 0.005835	0.976865 ± 0.000044	0.000	1
Zebra	2	0.827683 ± 0.000000	0.827683 ± 0.000000	0.827683 ± 0.000000	NaN	0
	3	0.905167 ± 0.000000	0.882824 ± 0.014098	0.905789 ± 0.000002	0.000	1
	4	0.939160 ± 0.000000	0.915819 ± 0.009666	0.939161 ± 0.000036	0.461	0
	5	0.956017 ± 0.000000	0.934314 ± 0.009491	0.955941 ± 0.000079	0.000	1

4.6. Running time analysis

Because real-time applications require a low running time in addition to high performance, the CPU times of the algorithms have been analysed. The mean and standard deviations of the CPU times of 30 runs have been reported in Table 13. The run time experiments were implemented in Delphi 7 and were executed on a work station with an Intel Core i7 2.67 GHz CPU and 8 GB of RAM. For all images, the mean overall CPU time is shown in Fig. 14. From the table and the figure, it can be seen that the algorithms have similar running times, and the running times change within the range of [0.04,0.06], which is reasonable for real-time applications. If a better time span is needed, optimised coding or parallel implementations of the algorithms can be used. The running times of the algorithms as a function of the problem dimensions has been illustrated in Fig. 15, which shows that the running times of the algorithms tend to grow at a linear rate as the problem dimension increases, for the dimensions considered. It can be concluded that the algorithms are scalable, which means that they

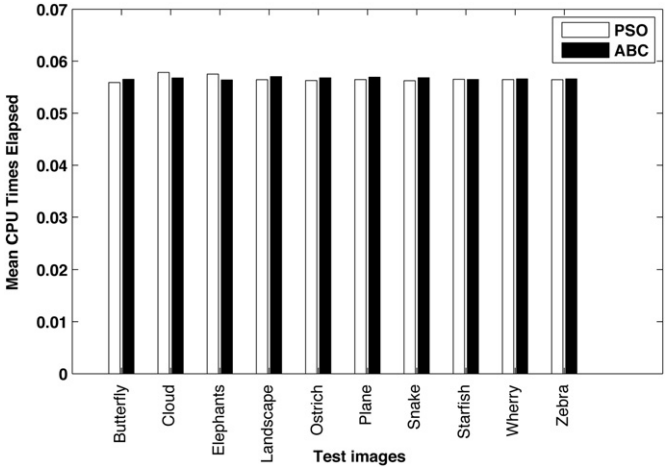


Fig. 14. The mean of the CPU times vs. the images.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 12
Statistical anaylsis of 30 runs for each of 48 cases for the experiments on Kapur's entropy criterion. p: probability of the statistic, $h = 0$ indicates null hypothesis cannot be rejected at the 0.05 significance level, $h = 1$ indicates null hypothesis can be rejected.

Image	m	Mean ± Std.Dev.		Wilcoxon rank sum	
		PSO	ABC	p	h
Butterfly	2	12.614205 ± 0.028321	12.650776 ± 0.000000	0.000	1
	3	15.665583 ± 0.053197	15.744889 ± 0.000290	0.000	1
	4	18.467064 ± 0.166572	18.814713 ± 0.000107	0.000	1
	5	20.987919 ± 0.214554	21.475912 ± 0.002039	0.000	1
Cameraman	2	12.125693 ± 0.049608	12.168753 ± 0.000000	0.000	1
	3	15.090287 ± 0.094854	15.227231 ± 0.000759	0.000	1
	4	17.957185 ± 0.246210	18.395514 ± 0.000114	0.000	1
	5	20.595292 ± 0.218829	21.139852 ± 0.004830	0.000	1
Cloud	2	12.672198 ± 0.006404	12.679798 ± 0.000000	0.000	1
	3	15.751573 ± 0.080614	15.851766 ± 0.000000	0.000	1
	4	18.490501 ± 0.111351	18.692038 ± 0.000539	0.000	1
	5	20.982239 ± 0.182849	21.365165 ± 0.002444	0.000	1
Elephants	2	12.096414 ± 0.021018	12.130062 ± 0.000000	0.000	1
	3	15.031706 ± 0.087136	15.192873 ± 0.000041	0.000	1
	4	17.692814 ± 0.132486	18.146334 ± 0.000095	0.000	1
	5	20.078257 ± 0.253976	20.792078 ± 0.002575	0.000	1
Landscape	2	11.064915 ± 0.023266	11.098550 ± 0.000000	0.000	1
	3	13.802983 ± 0.073615	14.012935 ± 0.002000	0.000	1
	4	16.224904 ± 0.204624	16.706681 ± 0.000832	0.000	1
	5	18.521024 ± 0.248820	19.209781 ± 0.021734	0.000	1
Lena512	2	12.320707 ± 0.024495	12.347015 ± 0.000000	0.000	1
	3	15.181341 ± 0.095991	15.318034 ± 0.000088	0.000	1
	4	17.691557 ± 0.256140	18.011257 ± 0.002147	0.000	1
	5	19.983394 ± 0.328139	20.608984 ± 0.001663	0.000	1
Ostrich	2	12.564730 ± 0.050872	12.612476 ± 0.000000	0.000	1
	3	15.539176 ± 0.093001	15.672139 ± 0.000100	0.000	1
	4	18.337238 ± 0.131413	18.610412 ± 0.006537	0.000	1
	5	20.943968 ± 0.216415	21.530876 ± 0.004892	0.000	1
Plane	2	11.103474 ± 0.070139	11.154947 ± 0.000000	0.000	1
	3	13.722223 ± 0.190416	14.040767 ± 0.000598	0.000	1
	4	16.032097 ± 0.318944	16.761368 ± 0.001925	0.000	1
	5	18.275825 ± 0.517449	19.225426 ± 0.003297	0.000	1
Snake	2	12.413651 ± 0.021528	12.434921 ± 0.000000	0.000	1
	3	15.614277 ± 0.065918	15.695427 ± 0.000009	0.000	1
	4	18.479917 ± 0.140132	18.711980 ± 0.000939	0.000	1
	5	21.061050 ± 0.225241	21.501349 ± 0.002179	0.000	1
Starfish	2	12.981312 ± 0.030787	12.999386 ± 0.000000	0.000	1
	3	16.097712 ± 0.066211	16.166475 ± 0.000048	0.000	1
	4	18.948067 ± 0.111675	19.117975 ± 0.000544	0.000	1
	5	21.483308 ± 0.232631	21.875180 ± 0.002011	0.000	1
Wherry	2	12.194804 ± 0.011229	12.204231 ± 0.000000	0.000	1
	3	15.111047 ± 0.057710	15.194558 ± 0.000160	0.000	1
	4	17.862073 ± 0.103027	18.107110 ± 0.000505	0.000	1
	5	20.465771 ± 0.175282	20.823483 ± 0.001706	0.000	1
Zebra	2	12.237921 ± 0.037634	12.272881 ± 0.000000	0.000	1
	3	15.185840 ± 0.066846	15.280784 ± 0.000083	0.000	1
	4	17.917022 ± 0.179291	18.151661 ± 0.000333	0.000	1
	5	20.347707 ± 0.172336	20.749142 ± 0.001903	0.000	1

are suitably efficient and practical in terms of time complexity for high-dimensional problems.

4.7. A discussion on PSO and ABC algorithms

In the experiments, two swarm-based algorithms have been employed: PSO and ABC. They are both simple models that can be easily applied to any kind of optimisation problem. Although the algorithms have been initialised by the same conditions and used the same objective function, they have very different search characteristics. The main properties of an optimisation algorithm are the exploration and exploitation processes. These processes are also referred as diversification and intensification. Exploration finds new solutions in search space and maintains diversity in the population, while exploitation tries to use the available knowledge in the population to improve the quality of the population. Evolutionary algorithms differ in how and when they do exploration and exploitation. The PSO algorithm uses two separate matrices: one for the current state of the population and one for holding the best state of each individual. When the algorithm makes a modification to the current position of an individual, it is directly accepted by the current population without checking if it is better or not than the current solution. This is the exploration property of the PSO algorithm, which allows new points in the search space to be added in the population. Although this may seem like the algorithm always does random fluctuations and maintains a lot of diversity, the modification of the solutions is not truly random. The solutions are updated using a directed search (Eq. (17)), which recruits the solutions to search using the information in the other matrix that holds the best states and is modified only if the new solution is better than its best solution. This is the exploitation property of the PSO algorithm.

The ABC algorithm performs exploration and exploitation in a different manner. The ABC algorithm has only one population holding the current state of the solutions. In the employed bee phase of the algorithm, each solution is exploited by combining the information of the solution itself and the information coming from a neighbour (19). The current solution is compared with

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

Table 13
Comparison of the mean and standard deviation of the CPU times (in seconds) elapsed by the PSO and ABC algorithms.

Image	m	Mean			
		PSO	ABC	PSO	ABC
Butterfly (481 × 321)	2	0.052933	0.052800	0.000019	0.000006
	3	0.054167	0.054867	0.000014	0.000039
	4	0.056700	0.057233	0.000019	0.000024
	5	0.059767	0.061200	0.000019	0.000026
Cameraman (256 × 256)	2	0.039000	0.038267	0.000032	0.000006
	3	0.040533	0.040667	0.000030	0.000020
	4	0.042833	0.043633	0.000037	0.000021
	5	0.043767	0.045567	0.000001	0.000019
Cloud (481 × 321)	2	0.053100	0.053800	0.000013	0.000035
	3	0.055067	0.055000	0.000024	0.000018
	4	0.062900	0.056900	0.000133	0.000003
	5	0.060267	0.061400	0.000027	0.000047
Elephants (481 × 321)	2	0.058067	0.052400	0.000107	0.000003
	3	0.054633	0.055233	0.000018	0.000018
	4	0.056800	0.057733	0.000002	0.000023
	5	0.060600	0.060200	0.000023	0.000023
Landscape (481 × 321)	2	0.053533	0.053400	0.000031	0.000019
	3	0.054967	0.055567	0.000032	0.000052
	4	0.057633	0.058133	0.000031	0.000031
	5	0.059633	0.061067	0.000018	0.000057
Lena (512 × 512)	2	0.041100	0.040200	0.000017	0.000016
	3	0.049333	0.043267	0.000144	0.000016
	4	0.045467	0.044933	0.000019	0.000018
	5	0.048000	0.048167	0.000048	0.000039
Ostrich (321 × 481)	2	0.053067	0.053100	0.000024	0.000017
	3	0.055167	0.054633	0.000038	0.000025
	4	0.056867	0.057800	0.000035	0.000017
	5	0.060000	0.061633	0.000006	0.000012
Plane (481 × 321)	2	0.053433	0.053700	0.000026	0.000032
	3	0.055400	0.055467	0.000017	0.000029
	4	0.057433	0.058233	0.000024	0.000022
	5	0.059567	0.060333	0.000041	0.000023
Snake (481 × 321)	2	0.053367	0.052900	0.000035	0.000020
	3	0.054300	0.055033	0.000024	0.000028
	4	0.057100	0.058133	0.000028	0.000026
	5	0.060167	0.061200	0.000026	0.000036
Starfish (481 × 321)	2	0.053067	0.053067	0.000022	0.000025
	3	0.054633	0.055267	0.000016	0.000083
	4	0.057967	0.057433	0.000041	0.000026
	5	0.060400	0.060233	0.000015	0.000020
Wherry (321 × 481)	2	0.053033	0.053233	0.000024	0.000036
	3	0.054700	0.054633	0.000013	0.000027
	4	0.056867	0.058133	0.000014	0.000026
	5	0.061267	0.060400	0.000035	0.000016
Zebra (481 × 321)	2	0.054000	0.052467	0.000039	0.000019
	3	0.054567	0.055100	0.000019	0.000031
	4	0.057000	0.057733	0.000026	0.000027
	5	0.060167	0.061100	0.000032	0.000016

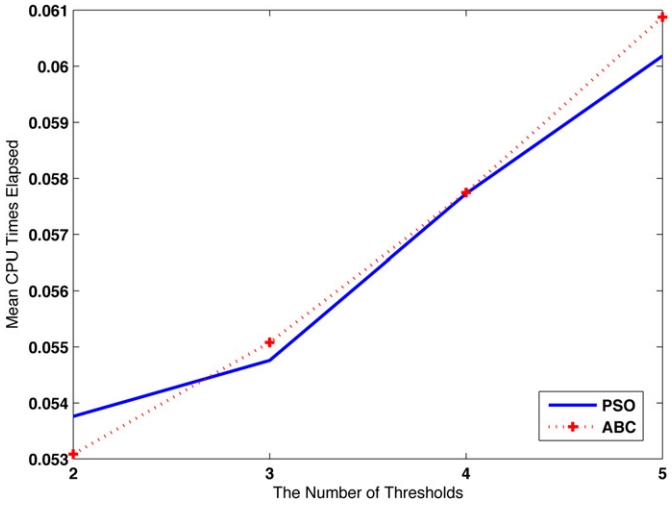


Fig. 15. The mean of the CPU times vs. the number of the thresholds (m).

its mutant, and it is replaced with the mutant provided that the mutant is better. This is the exploitation property of ABC. In the onlooker bee phase of the algorithm, the algorithm tends to search the neighbourhood of solutions sampled by a stochastic roulette-wheel selection, which directs the search to the vicinity of the solutions with high probability and allows poor solutions to be intensified. The solutions that have better mutants are discarded, and their mutants are retained in the population. This is another exploitation based on stochastic sampling. In both the employed bee and onlooker bee phases, if the solution cannot be improved by its mutant, the counter associated with that solution is incremented by one. In the scout bee phase, the counters are checked to determine the solutions that could not be improved after a predefined number of trials. The determined solutions are replaced with new random solutions in the search space. This is the exploration property of the ABC algorithm.

As mentioned above, two algorithms can use different search operators and selection operators to conduct exploration and exploitation. A good optimisation algorithm explores the search space and then exploits the current state efficiently in a balanced

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072

manner. The experimental results suggest that the ABC algorithm has better search characteristics under multilevel thresholding despite its simplicity. Another advantage of the ABC algorithm is that is has only one control parameter apart from the population size and the iteration number, which are common for all population-based algorithms.

5. Conclusion

In this study, Otsu's method and the PSO and ABC algorithms are employed to maximise the between-class variance and Kapur's entropy separately to find multilevel thresholds. The results evaluated by statistical tools suggest that the ABC algorithm with both the between-class variance and the entropy criterion can be efficiently used in multilevel thresholding. When the number of thresholds is two, the ABC algorithm and the PSO algorithm can be used as an alternative to Otsu's method. On the images that have histograms locally located in some grey levels rather than having a regular distribution, entropy-based segmentation gives poor results, while on the others, the ABC algorithm with Kapur's entropy criterion emphasises the separability and compactness of the groups efficiently, due to a good balance between global search and local intensification operators. Another advantage of the ABC algorithm is that it has fewer control parameters than PSO. The results were also evaluated in terms of PSNR and SSIM. Kapur's entropy and the between-class variance work on the relative frequencies of the grey levels in each group; therefore, different metrics, such as PSNR based on the mean square error, may not yield the same output. Experiments on the running times of the algorithms show that both algorithms are scalable and that the running times of the algorithms grow at a linear rate with respect to the size of the problem.

References

[10] P.D. Sathya, R. Kayalvizhi, Modified bacterial foraging algorithm based multilevel thresholding for image segmentation, Expert Systems with Applications 24 (2011) 595–615.
[11] Du Feng, Shi Wenkang, Chen Liangzhou, Deng Yong, Zhu Zhenfu, Infrared image segmentation with 2-D maximum entropy method based on particle swarm optimization (PSO), Pattern Recognition Letters 26 (2005) 597–603.
[12] P-S. Liao, P-C. Chung, T-S. Chen, A fast algorithm for multilevel thresholding, Journal of Information Science and Engineering 17 (2001) 713–727.
[13] J.C. Park, S. Abusalah, Maximum entropy: a special case of minimum cross-entropy applied to nonlinear estimation by an artificial neural network, Complex Systems 11 (1997) 289–307.
[14] J.N. Kapur, Maximum-entropy Models in Science and Engineering, John Wiley and Sons, 1989.
[15] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging 13 (1) (2004) 146–165.
[16] M.-H. Horng, Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization, Expert Systems with Applications 37 (2010) 4580–4592.
[17] Y. Zhiwei, Z. Zhaobao, Y. Xin, N. Xiaogang, Automatic threshold selection based on ant colony optimization algorithm, in: Proceedings of the International Conference on Neural Networks and Brain, Number 728–732, Beijing, 2006.
[18] P-Y. Yin, A fast scheme for multilevel thresholding using genetic algorithms, Signal Processing 72 (1999) 85–95.
[19] P.Y. Yin, Multilevel minimum cross entropy threshold selection based on particle swarm optimization, Applied Mathematics and Computation 184 (2) (2007) 503–513.
[20] M. Maitra, A. Chatterjee, A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding, Expert Systems with Applications 34 (2008) 1341–1350.
[21] C. Guo, H. Li, Multilevel thresholding method for image segmentation based on an adaptive particle swarm optimization algorithm, AI 2007, LNAI 4830 (2007) 654–658.
[22] P.D. Sathya, R. Kayalvizhi, Optimal multilevel thresholding using bacterial foraging algorithm, Expert Systems with Applications 38 (12) (2011) 15549–15564.
[23] P.D. Sathya, R. Kayalvizhi, Optimal segmentation of brain MRI based on adaptive bacterial foraging algorithm, Neurocomputing 74 (14–15) (2011) 2299–2313.
[24] M.-H. Horng, T.-W. Jiang, Multilevel image thresholding selection using the artificial bee colony algorithm., in: Artificial Intelligence and Computational Intelligence, Lecture Notes in Computer Science, vol. 6320/2010, 2010, pp. 318–325.
[25] Y. Zhang, L. Wu, Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach, Entropy 13 (2011) 841–859.
[26] J. Kennedy, R.C. Eberhart, Particle swarm optimization., in: 1995 IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[27] R. Poli, Analysis of the publications on the applications of particle swarm optimisation, Journal of Artificial Evolution and Applications 2008 (2008) 1–10.
[28] F. Valdez, P. Melin, O. Castillo, An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, Applied Soft Computing 11 (2) (2011) 2625–2632.
[29] F. Al-Obeidat, N. Belacel, J.A. Carretero, P. Mahanti, An evolutionary framework using particle swarm optimization for classification method proaftn, Applied Soft Computing 11 (8) (2011) 4971–4980.
[30] A.W. Mohemmed, N.C. Sahoo, T.K. Geok, Solving shortest path problem using particle swarm optimization, Applied Soft Computing 8 (4) (2008) 1643–1653.
[31] R.J. Kuo, C.M. Chao, Y.T. Chiu, Application of particle swarm optimization to association rule mining, Applied Soft Computing 11 (1) (2011) 326–336.
[32] I.J. Raglend, C. Raghuveer, G.R. Avinash, N.P. Padhy, D.P. Kothari, Solution to profit based unit commitment problem using particle swarm optimization, Applied Soft Computing 10 (4) (2010) 1247–1256.
[33] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[34] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Applied Soft Computing 8 (1) (2008) 687–697.
[35] B. Akay, D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, Information Sciences 192 (2012) 120–142.
[36] D. Karaboga, B. Akay, A survey: algorithms simulating bee swarm intelligence, Artificial Intelligence Review 31 (1) (2009) 55–68.
[37] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing 9 (2) (2009) 625–631.
[38] M. Sonmez, Artificial bee colony algorithm for optimization of truss structures, Applied Soft Computing 11 (2) (2011) 2406–2418.
[39] T-J. Hsieh, H-F. Hsiao, W-C. Yeh, Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm, Applied Soft Computing 11 (2) (2011) 2510–2525.
[40] D. Karaboga, B. Akay, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, Applied Soft Computing 11 (3) (2011) 3021–3031.
[41] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation 214 (2008) 108–132.
[42] N. Otsu, A threshold selection method from gray-level histograms, IEEE Transactions on Systems, Man and Cybernetics 9 (1) (1979) 62–66.
[43] P.K. Sahoo, J.N. Kapur, A.K.C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Computer Vision Graphics Image Processing 29 (1985) 273–285.
[44] M. Portes de Albuquerque, I.A. Esquef, A.R. Gesualdi Mello, Image thresholding using tsallis entropy, Pattern Recognition Letters 25 (2004) 1059–1065.
[48] C. Grosan, A. Abraham, Stigmergic Optimization (Studies in Computational Intelligence) Chapter Stigmergic Optimization: Inspiration Technologies and Perspectives, Springer-Verlag New York, Inc, 2006, pp. 1–24.
[49] 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: In Proc. 8th Int'l Conf. Computer Vision, vol. 2, July, 2001, pp. 416–423.
[50] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error measurement to structural similarity, IEEE Transactions on Image Processing 13 (1) (2004) 600–612.

Please cite this article in press as: B. Akay, A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding, Appl. Soft Comput. J. (2012), http://dx.doi.org/10.1016/j.asoc.2012.03.072