



# Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation

Ming-Huwi Horng\*

Department of Computer Science and Information Engineering, National PingTung Institute of Commerce, No. 51 Min-Shien E. Road, Pingtung City 900, Taiwan

## ARTICLE INFO

### Keywords:

Particle swarm optimization  
Honey bee mating optimization  
Hybrid cooperative-comprehensive learning based PSO algorithm  
Fast Otsu's method  
Artificial bee colony algorithm

## ABSTRACT

Multilevel thresholding is an important technique for image processing and pattern recognition. The maximum entropy thresholding (MET) has been widely applied in the literature. In this paper, a new multilevel MET algorithm based on the technology of the artificial bee colony (ABC) algorithm is proposed: the maximum entropy based artificial bee colony thresholding (MEABCT) method. Four different methods are compared to this proposed method: the particle swarm optimization (PSO), the hybrid cooperative-comprehensive learning based PSO algorithm (HCOCLPSO), the Fast Otsu's method and the honey bee mating optimization (HBMO). The experimental results demonstrate that the proposed MEABCT algorithm can search for multiple thresholds which are very close to the optimal ones examined by the exhaustive search method. Compared to the other four thresholding methods, the segmentation results of using the MEABCT algorithm is the most, however, the computation time by using the MEABCT algorithm is shorter than that of the other four methods.

© 2011 Elsevier Ltd. All rights reserved.

## 1. Introduction

Thresholding is one of the most important techniques for performing image segmentation. It is generally simple and computationally efficient. The main objective is to determine a threshold for bi-level thresholding or several thresholds for multilevel thresholding for image segmentation. Bi-level thresholding selects only one threshold which separates the pixels into two classes; multilevel thresholding determines multiple thresholds which divide pixels into several groups. In general, the global thresholding methods can be classified as parametric and nonparametric. These methods select thresholds by optimizing (maximizing or minimizing) some criterion functions defined from images. In the parametric approaches, the gray-level distribution of each class has a probability density function that is generally assumed to obey a Gaussian distribution. These methods of parametric approaches attempt to estimate of the parameters of distribution that will best fit the given histogram data. It typically leads to a nonlinear optimization problems of which the solution is computationally expensive and time-consuming. Kittler and Illingworth (1986) proposed a thresholding method that approximates the histogram by a mixture of normal distributions and minimizes the classification error probability. Wang, Chung, and Xiong (2008) proposed a method which is rooted in the Parzen window estimate of the unknown gray value probability density function. The method can

integrate image histogram information with the spatial information about pixels of different gray levels. Zahara, Fan, and Tsai (2005) presented a hybrid optimization scheme for multiple thresholding by the criteria of Otsu's minimum within-class variance and Gaussian function fitting.

Nonparametric approaches find the thresholds that separate the gray-level regions of an image in an optimal manner based on some discriminating criteria such as the between-class variance, entropy and cross entropy. The popular method, Otsu's method (1979), selected optimal thresholds by maximizing the between-class variance. Shoo, Soltani, Wong, and Chen (1988) found that the Otsu's method is one of the better threshold selection methods for real world images with regard to uniformity and shape measures. However, inefficient formulation of between-class variance makes the methods very time consuming in multilevel threshold selection. To solve this problem, Liao, Chen, and Chung (2001) proposed a fast recursive algorithm, Fast Otsu's method, along with a look-up-table to implement in the application of multilevel thresholding. Ye, Chen, Li, and Zhang (2008) proposed a particle swarm optimization (PSO) algorithm to optimize the Otsu's criterion. Kapur, Sahoo, and Wong (1985) proposed a method for gray-level picture thresholding using the entropy of the histogram. Abutaleb (1989) proposed a 2-D maximum entropy thresholding method for separating the regions of image. Zhang and Liu (2006) adopted the particle swarm optimization algorithm to maximize the entropy for underwater image segmentation. Madhubanti and Amitava (2008) proposed a hybrid cooperative-comprehensive learning based PSO algorithm (HCOLPSO) based on maximum entropy

\* Tel./fax: +886 8 7238700.

E-mail address: [horng@npic.edu.tw](mailto:horng@npic.edu.tw)

criterion. Li and Lee (1993) proposed a method which selects the threshold by minimizing the cross entropy between the original and segmented images. Yin (2007) developed a recursive programming techniques to reduce the order of magnitude of computing the multilevel thresholds and further used the PSO algorithm to minimize the cross entropy. Horng (2010) applied the honey bee mating optimization (HBMO) to search for the thresholds of histogram of image. The developed method was called maximum entropy honey bee mating optimization (MEHBOT) algorithm. The experimental results demonstrated that the result of the MEHBOT algorithm was superior to other algorithms such as the PSO, HCOL-PSO and Fast Otsu's methods. Hammouche, Diag, and Siarry (2010) compared various meta-heuristic techniques implemented in the multilevel thresholding. They found that the differential evolution was the most efficient and the particle swarm optimization converged the most quickly.

Over the last decade, modeling the behavior of social insects, such as ants and bees, for the purpose of search and problems solving has been the context of the emerging area of swarm intelligence. The artificial bee colony (ABC) algorithm may also be considered as a typical swarm-based approach for optimization, in which the search algorithm is inspired by the foraging behavior of bee colonies. The search algorithm has led to the emergence of collective intelligence of bee swarms consisting of three essential components: food source, employed foragers and unemployed foragers. There are two leading modes of the bee colony behavior: recruitment to and abandonment of a food source. The unemployed foragers are onlookers and scouts. Karaboga and Basturk (2007) have recently proposed a developed model of the artificial bee colony (ABC) algorithm that simulated these social behaviors of honey bees for searching for the numerical optimization problems. Singh (2009) applied the ABC algorithm, called the ABC-LCMST algorithm, for the leaf-constrained minimum spanning tree problem. Computational results demonstrated the superiority of the new algorithm over other approaches such as ant colony algorithm and the tabu search. The performance of the ABC algorithm on clustering is also compared with set of results of clustering methods that are given in the work of Karaboga and Ozturk (2011).

This paper applies the ABC algorithm to search for the multilevel thresholds using the maximum entropy (MET) criterion. This proposed method is called the maximum entropy based artificial bee colony thresholding (MEABCT) algorithm. In the experiments presented in this paper, the exhaustive search method is conducted to derive the optimal solutions for comparison with the results generated from MEABCT algorithm. The four different methods – the PSO, the hybrid cooperative-comprehensive learning based PSO algorithm (HCOCLPSO), the Fast Otsu's method, and the MEHBOT – are implemented in the several real images for purposes of comparison.

The remainder of the paper is organized as follows: Section 2 introduces the artificial bee colony algorithm. Section 3 presents the proposed MEABCT algorithm. Performance evaluation and experimental results are presented in detail in Section 4. Finally, some conclusions are made in Section 5.

## 2. Artificial bee colony algorithm

The colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. The employed bees bring loads of nectar from the food resource to the hive and may share the information about food source in the dancing area. These bees carry information about food sources and share them with a certain probability by dancing in a dancing area in the hive. The onlooker bees wait in the dances area for making a decision on the selection of a food source depending on the probability delivered by employed

bees. The computation of probability is based on the amounts of the food source. The other kind of bee is scout bee that carries out random searches for new food sources. The employed bee of an abandoned food source becomes a scout and as soon as it finds a new food source it becomes employed again. In other words, the each search cycle of the ABC algorithm contains three steps. First, the employed bees are sent into their food sources and the amounts of nectar are evaluated. After sharing this information about the nectar, onlooker bees select the food source regions and evaluating the amount of nectar in the food sources. The scout bees and then chosen and sent out to find the new food sources.

The artificial bee colony (ABC) algorithm was proposed by Karaboga and Basturk (2008). In this algorithm, the position of a food source  $z_i$  represents a possible solution to the optimization problems and the amount of nectar in a food source corresponds to the fitness  $fit(z_i)$  of the corresponding solution. The number of employed or onlooker bees is generally equal to the number of solutions in the population of solutions. Initially, the ABC algorithm randomly produced a distributed initial population  $P$  of  $SN$  solutions, where  $SN$  denoted the number of employed bees or onlooker bees. Each solution  $z_i$  ( $i = 1, 2, \dots, SN$ ) is a  $D$ -dimensional vector. Here  $D$  is the number of optimization parameters. In each execution cycle,  $C$  ( $C = 1, 2, \dots, MCN$ ), the population of the solutions is subjected to the search processes of the employed, the onlooker and scout bees. An employed bee modifies the possible solution depending on the amount of nectar (fitness value) of the new source (new solution) by using Eq. (1)

$$v_{ij} = z_{ij} + \phi_{ij}(z_{ij} - z_{kj}) \quad (1)$$

where  $k \in \{1, 2, \dots, SN\}$  but  $k \neq i$  and  $j \in \{1, 2, \dots, D\}$  are randomly selected indexes.  $\phi_{ij}$  is a random number between  $[-1, 1]$ .

If there is more nectar in new solution is than that in the previous one, the bee remembers the new position and forgets the old one, otherwise it retains the location of the previous one. When all employed bees have finished this search process, they deliver the nectar information and the position of the food sources to the onlooker bees, each of whom chooses a food source according to a probability proportional to the amount of nectar in that food source. The probability  $p_i$  of selecting a food source  $z_i$  is determined using the following Eq. (2)

$$p_i = \frac{fit(z_i)}{\sum_{i=1}^{SN} fit(z_i)} \quad (2)$$

In practical terms, any food source  $z_i$  ( $i = 1, 2, \dots, SN$ ) sequentially generates a random number between  $[0, 1]$  and if this number is less than  $p_i$ , an onlooker bee are sent to food source  $z_i$  and produces a new solution based on Eq. (1). If the fitness of the new solution is more than the old one, the onlooker memorizes the new solution and shares this information with other onlooker bees. Otherwise, the new solution will be discarded. The process is repeated until all onlookers have been distributed to the food sources and produces the corresponding new solution.

If the position of food source can not be improved through the predetermined number of "limit" of bees, then the food resource  $z_i$  is abandoned and then the employed bee becomes a scout. Assume that the abandoned source is  $z_i$  and  $j \in \{1, 2, \dots, D\}$ , then the scout discovers a new food source to be replaced with  $z_i$ . This operation can be defined as in (3)

$$z_{ij} = z_{min}^j + rand(0, 1)(z_{max}^j - z_{min}^j) \quad (3)$$

where the  $z_{min}^j$  and  $z_{max}^j$  are the upper bound and upper bound of the  $j$ th component of all solutions. If the new solution is better than the abandoned one, the scout will become an employed bee. The selection of employed bees, onlooker bees and scouts is repeated until the termination criteria have been satisfied.

**Table 1**

Parameter set for the MEHB MOT algorithm.

Parameters	Values
Number of queens	1
Number of drones (m)	5
Number of iterations	100
Speed reduction schema ( $\alpha$ )	0.98
Capacity of spermatheca ( $n_{sperm}$ )	20
Speed of queen at first of flight	1.0
The breeding ratio ( $P_c$ )	0.8
Mutation ratio ( $P_m$ )	0.01
Mutation variation ( $e$ )	0.5

**Table 2**

Parameter set for the PSO algorithm.

Parameters	Values
Number of particles ( $S$ )	50
Velocities randomly ( $V$ )	$[-1.0, 1.0]$
Number of iterations (iter)	100
Cognitive coefficient ( $C1$ )	2.1
Cognitive coefficient ( $C2$ )	2.0
$\Phi = C1 + C2$	4.1
$K = \frac{2}{ 2 - \phi - \sqrt{\phi^2 - 4\phi} }$	0.729
Initialization range for the position of the particles $[f_{min}, f_{max}]$	$[0, 255]$

**Table 3**

Parameter set for the HCOCLPSO algorithm.

Parameters	Values
Number of particles	50
Number of iterations	100
Initial inertia weight ( $\omega_{initial}$ )	0.9
Slope of inertia weight ( $m_{1w}$ )	$2.5 \times 10^{-4}$
$c'_i$	1.49445
Initialization range for the position of the particles $[f_{min}, f_{max}]$	$[0, 255]$
Selection probability ( $P_c$ )	0.2

### 3. Proposed approach

#### 3.1. Entropy criterion based measure

The entropy criterion, proposed by Kapur et al. (1985), was widely used in determining the optimal thresholding in image segmentation. The original algorithm had been developed for bi-level thresholding. The method can also extend to solve multilevel thresholding problems and can be described as follows.

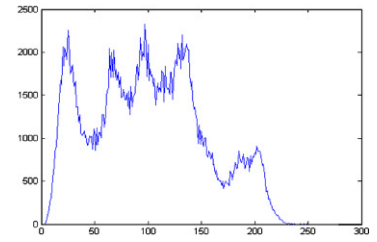
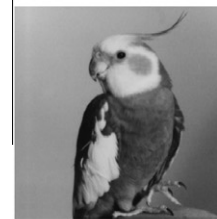
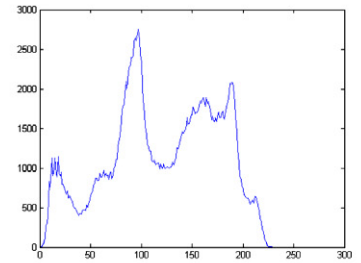
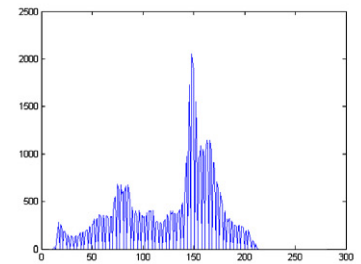
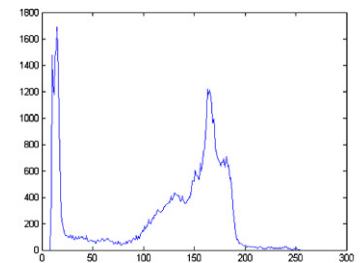
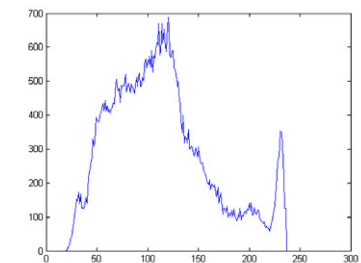
Let there be  $L$  gray levels in a given image  $I$  and these gray levels are in the range  $\{0, 1, 2, \dots, L-1\}$ . Then one can define  $P_i = h(i)/N$ , ( $0 \leq i \leq L-1$ ) where

$h(i)$  denotes the number of pixels with gray-level  $i$ .  
 $N$  denotes total number of pixels in the image.

Here, given a problem to select  $D$  thresholds,  $[t_1, t_2, \dots, t_D]$  for a given image  $I$ , the objective function  $f$  is to maximize:

$$f([t_1, t_2, \dots, t_D]) = H_0 + H_1 + H_2 + \dots + H_D \quad (4)$$

$$\begin{aligned} \omega_0 &= \sum_{i=0}^{t_1-1} P_i, & H_0 &= - \sum_{i=0}^{t_1-1} \frac{P_i}{\omega_0} \ln \frac{P_i}{\omega_0} \\ \omega_1 &= \sum_{i=t_1}^{t_2-1} P_i, & H_1 &= - \sum_{i=t_1}^{t_2-1} \frac{P_i}{\omega_1} \ln \frac{P_i}{\omega_1} \\ \omega_2 &= \sum_{i=t_2}^{t_3-1} P_i, & H_2 &= - \sum_{i=t_2}^{t_3-1} \frac{P_i}{\omega_2} \ln \frac{P_i}{\omega_2}, \dots \\ \omega_D &= \sum_{i=t_D}^{255} P_i, & H_D &= - \sum_{i=t_D}^{255} \frac{P_i}{\omega_D} \ln \frac{P_i}{\omega_D} \end{aligned}$$

(a) LENA ( $512 \times 512$ )(b) PEPPER ( $512 \times 512$ )(c) BIRD ( $256 \times 256$ )(d) CAMERA ( $256 \times 256$ )(e) GOLDHILL ( $256 \times 256$ )**Fig. 1.** The test images and corresponding histograms.

In our proposed HEABCT algorithm, we try to obtain this optimum  $D$ -dimensional vector  $[t_1, t_2, \dots, t_D]$ , which can maximize (4). The objective function is also used as the fitness function of the HEABCT algorithm.

**Table 4**

The selection thresholds for five test images by using the MEABCT algorithm and the exhaustive search method ( $k$ : number of thresholds).

Image (size)	$k$	Exhaustive		MEABCT	
		Thresholds	Computation time (ms)	Thresholds	Computation time (ms)
LENA (512 × 512)	2	80,150	4.89	80,150	1.39
	3	60,109,160	158.49	60,109,160	5.94
	4	56,100,144,182	8290	56,100,144,182	24.39
	5	44,79,114,150,185	451304	44,79,115,148,186	189.35
PEPPER (512 × 512)	2	74,146	3.73	74,146	1.45
	3	61,112,164	145.58	61,112,164	5.98
	4	57,104,148,194	7965	57,104,148,194	29.78
	5	42,77,113,153,194	439784	42,77,113,153,194	187.35
BIRD (256 × 256)	2	71,138	4.13	71,138	1.12
	3	70,129,177	132.67	70,129,177	2.89
	4	51,96,139,177	6564	51,94,138,178	17.85
	5	46,74,104,141,177	414789	46,74,105,142,177	109.35
CAMERA (256 × 256)	2	128,193	4.54	128,193	1.28
	3	44,104,193	138.67	44,104,193	5.12
	4	44,97,146,197	7169	44,97,146,197	22.58
	5	40,84,119,155,197	439697	40,83,119,154,197	98.26
GOLDHILL (256 × 256)	2	90,157	5.57	90,157	1.39
	3	79,132,178	140.17	79,132,178	5.16
	4	67,108,151,191	7190	67,108,151,191	27.25
	5	61,96,132,166,199	447429	59,96,131,167,199	124.69

**Table 5**

Thresholds, computation times, PSNR values and Fitness values for test images by using MEABCT algorithm.

Image	Number of thresholds	Thresholds	Computation time (ms)	PSNR (dB)	Fitness value
LENA (512 × 512)	2	80,150	1.39	15.46	12.6990
	3	60,109,160	5.94	18.55	15.7658
	4	56,100,144,182	24.39	19.71	18.5875
	5	44,79,115,148,186	189.35	21.68	21.2468
PEPPER (512 × 512)	2	74,146	1.45	16.47	12.6348
	3	61,112,164	5.98	18.42	15.6892
	4	57,104,148,194	29.78	19.21	18.5397
	5	42,77,113,153,194	187.35	21.81	21.2830
BIRD (256 × 256)	2	71,138	1.12	17.44	11.1647
	3	70,129,177	2.89	18.53	13.8659
	4	51,94,138,178	17.85	20.84	16.4558
	5	46,74,105,142,177	109.35	22.72	18.6961
CAMERA (256 × 256)	2	128,193	1.28	13.65	12.1688
	3	44,104,193	5.12	14.61	15.2274
	4	44,97,146,197	22.58	20.21	18.3995
	5	40,83,119,154,197	98.26	22.59	21.0831
GOLDHILL (256 × 256)	2	90,157	1.39	14.26	12.5384
	3	79,132,178	5.16	16.05	15.5957
	4	67,108,151,191	27.25	18.60	18.3957
	5	59,96,131,167,199	124.69	20.96	21.0934

### 3.2. Maximum entropy artificial bee colony thresholding (MEABCT) algorithm

In this paper, a maximum entropy based artificial bee colony thresholding (MEABCT) algorithm is developed based on the meta-heuristic approach proposed by Karaboga (2010). The details of MEABCT algorithm are introduced as follows.

#### 3.2.1. Step 1. Generate the initial population of solutions

Generate the  $SN$  solutions  $z_i$  ( $i = 1, 2, \dots, SN$ ) with  $D$  dimensions denoted by matrix  $Z$

$$Z = [z_1, z_2, \dots, z_{SN}],$$

$$z_i = (z_{i,1}, z_{i,2}, \dots, z_{i,D}) \quad (5)$$

where  $z_{ij}$  is the  $j$ th component value that is restricted into  $[0, \dots, L]$  and the  $z_{ij} < z_{i,j+1}$  for all  $j$ . The fitness of all solutions  $z_i$  is evaluated

and then set cycle = 1 and the trail number of each solution  $z_i$ ,  $trail_i$ , is equal to 0.

#### 3.2.2. Step 2. Place the employed bees on their food sources

In Step 2, each employed bee produces a new solution  $v_i$  by using (1) and computes the fitness value of the new solution. If the fitness of the new one is higher than that of the previous one, the employed bee memorizes the new position and forgets the old one; otherwise the employed bee keeps the old solution.

#### 3.2.3. Step 3. Send the onlooker bees to the food sources depending on their amount of nectar

In Step 3, we first calculate the probability value  $p_i$  of the solution  $z_i$  by means of their fitness values using (2). An onlooker bee selects a solution to update its solution depending on the probabilities and determines a neighbor solution around the chosen one. In the selection procedure for the first onlooker, a random number is

produced between  $[0, 1]$  and if this number is less than  $p_1$ , the solution is updated. Otherwise, the random number is compared with  $p_2$  and if less than that, the second solution is chosen. Otherwise, the third probability of third solution is checked. This process is repeated until all onlookers have been distributed to solutions. The distributed onlooker bee updates its own solution just as the employed bees do.

### 3.2.4. Step 4. Send the scouts to the search area to discover new food sources

If the solution  $z_i$  is not improved through Steps 2 and 3, the  $trail_i$  value of solution  $z_i$  will be increased by 1. If the  $trail_i$  of solution is more than the predetermined “limit” the solution  $z_i$  is considered to be an abandoned solution, meanwhile, the employed bee will be changed into a scout. The scout randomly produces the new solution by (6) and then compares the fitness of new solution with that of its old one. If the new solution is better than the old solution, it is replaced with the old one and set its own  $trail_i$  into 0. This scout will be changed into an employed bee. Otherwise, the old one is retained in the memory

$$Z_{ij} = z_{\min,j} + \text{rand}(0, 1)(z_{\max,j} - z_{\min,j}), \quad j = 1, 2, \dots, D \quad (6)$$

where the  $z_{\min,j}$  and  $z_{\max,j}$  are the minimum and maximum of  $j$ th component of all solutions, the  $\text{rand}(0, 1)$  is a random number generating function that generates the random number between  $[0, 1]$ .

### 3.2.5. Step 5. Record the best solution

In this step, the best solution so far is recorded and increases the cycle by 1.

### 3.2.6. Step 6. Check the termination criterion

If the cycle is equal to the maximum cycle number (MCN) then the algorithm is finished; otherwise go to Step 2.

## 4. Experimental results and comparative performances

We implement the all of the algorithms in Visual C++ 6.0 on a personal computer with 2.4 GHz CPU, 1G RAM running window XP system. The designed programs are revised from ones given by the homepage of artificial bee colony algorithm (Karaboga,

2010). Three control parameters that are the number of food sources which is equal to the number of employed bees or onlooker bees ( $SN$ ), the value of “limit” and the maximum cycle number (MCN) are set in the colony size ( $SN$ ) 30, MCN 100, and limit value 100. Tables 1–3 show that the parameters of the other multilevel thresholding algorithms. Five images named “LENA” “PEPPER” “BIRD” “CAMERA” and “GOLDHILL” are used for conducting our experiments. These original test images and their histograms are shown in Fig. 1.

The popular performance indicator, peak signal to noise ratio (PSNR), is used to compare the segmentation results by using the multilevel image threshold techniques (Yin, 2007). For the sake of completeness we define PSNR, measured in decibel (dB) as

$$\text{PSNR} = 20 \log_{10} \left( \frac{255}{\text{RMSE}} \right) \quad (\text{dB}) \quad (10)$$

where RMSE is the root mean-squared error, defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^M \sum_{j=1}^N (I(i,j) - \hat{I}(i,j))^2}{MN}} \quad (11)$$

Here  $I$  and  $\hat{I}$  are original and segmented images of size  $M \times N$ , respectively.

First, we execute the MEABCT algorithm on partitioning the five test images. The exhaustive search is also conducted for deriving the optimal solution for comparison. Table 4 shows the selected thresholds derived by the MEABCT algorithm and the optimal thresholds generated from the exhaustive search method. We find that the selected thresholds of MEABCT algorithm are equivalent or very close to optimal thresholds derived by the exhaustive search methods. Furthermore, we find that the computation times of exhaustive search method grows exponentially with the number of required thresholds. Obviously, the computation needs for the exhaustive search are absolutely unacceptable for  $T \geq 4$  ( $T$ : number of thresholds). The computation times of the MEABCT algorithm is significantly faster compared to the exhaustive search algorithm.

For evaluating the performance of the proposed MEABCT algorithm, we have implemented this method on the five test images. The performance metrics for checking the effectiveness of the method are chosen as the computation time so as to get

**Table 6**  
Thresholds, computation times, PSNR values and fitness values for test images by using MEABCT algorithm.

Image	$k$	Selected thresholds				
		MEABCT	MEABMOT	PSO	HCOCLPSO	Fast Otsu's method
LENA (512 × 512)	2	80,150	80,150	80,150	80,150	77,145
	3	60,109,160	60,109,160	60,109,160	60,109,160	56,106,159
	4	56,100,144,182	56,100,144,182	56,100,144,182	56,100,144,182	74,112,144,179
	5	44,79,115,148,186	44,80,115,150,185	43,79,114,150,185	46,83,118,153,187	45,79,109,138,173
PEPPER (512 × 512)	2	74,146	74,146	74,146	74,146	67,134
	3	61,112,164	61,112,164	72,135,193	61,112,164	61,117,165
	4	57,104,148,194	57,104,148,194	58,105,148,194	57,104,148,194	46,85,125,168
	5	42,77,113,153,194	42,77,113,153,194	43,77,113,153,194	42,77,114,154,194	41,77,111,145,176
BIRD (256 × 256)	2	71,138	71,138	71,138	71,138	68,124
	3	70,129,177	70,129,177	70,129,177	70,130,177	65,116,159
	4	51,96,139,177	51,96,139,177	51,94,138,177	51,96,140,177	58,96,131,163
	5	46,74,104,141,177	46,74,104,141,177	51,96,139,177,248	44,71,97,139,177	57,93,128,155,177
CAMERA (256 × 256)	2	128,193	128,193	128,193	128,193	69,143
	3	44,104,193	44,104,193	44,104,193	44,104,193	58,118,155
	4	44,97,146,197	44,97,146,197	44,97,146,197	44,97,146,197	41,94,139,169
	5	40,84,119,155,197	40,84,119,155,197	40,85,120,155,197	40,84,119,155,197	35,81,121,148,172
GOLDHILL (256 × 256)	2	90,157	90,157	90,157	90,157	93,160
	3	79,132,178	79,132,178	79,132,178	79,132,178	82,125,178
	4	67,108,151,191	67,108,151,191	66,107,151,191	65,104,144,186	69,102,137,185
	5	61,96,132,166,199	61,96,132,166,199	58,94,132,166,199	61,96,132,166,199	62,90,116,146,190



**Table 7**

The computation times and the corresponding PSNR of the five different multilevel thresholding methods.

Image	<i>k</i>	Computation times (ms)/PSNR (dB) ( <i>k</i> : number of thresholds)									
		MEABCT		MEHBMOT		PSO		HCOCLPSO		Fast Otsu's method	
LENA (512 × 512)	2	1.39	15.46	1.45	15.46	1.36	15.46	1.69	15.46	1.95	15.57
	3	5.94	18.55	6.95	18.55	4.89	18.55	13.58	18.55	32.68	18.55
	4	24.39	19.71	23.65	19.71	25.69	19.71	169.5	19.71	1968	20.57
	5	189.35	21.68	432.6	21.63	137.56	21.61	1158	21.56	18455	21.76
PEPPER (512 × 512)	2	1.45	16.47	1.87	16.47	1.56	16.47	2.26	16.47	1.96	16.27
	3	5.98	18.42	6.78	18.42	5.23	17.40	18.43	18.42	54.34	18.47
	4	29.78	19.21	36.76	19.21	28.43	19.23	219.6	19.21	2319	20.70
	5	187.35	21.81	234.9	21.81	154.26	21.39	1086	21.41	16672	22.36
BIRD (256 × 256)	2	1.12	17.44	1.09	17.44	1.15	17.44	2.10	17.44	2.32	16.65
	3	2.89	18.53	3.94	18.53	3.17	18.23	15.28	18.34	45.73	18.82
	4	17.85	20.84	18.65	20.77	19.94	20.73	132.5	20.89	204.39	20.97
	5	109.35	22.72	106.1	22.65	113.97	20.77	1153	22.20	13672	21.90
CAMERA (256 × 256)	2	1.28	13.65	1.95	13.65	1.34	13.65	2.25	13.65	3.58	17.20
	3	5.12	14.61	4.25	14.61	4.56	14.61	16.59	14.61	36.54	20.23
	4	22.58	20.21	24.69	20.21	23.56	20.21	112.5	20.21	352.2	21.53
	5	98.26	22.59	159.2	22.53	127.43	22.38	925.5	22.33	15594	23.28
GOLDHILL (256 × 256)	2	1.39	14.26	1.29	14.26	1.43	14.26	2.26	14.26	3.58	14.05
	3	5.16	16.05	6.52	16.05	5.12	16.05	19.25	16.05	36.54	15.55
	4	27.25	18.60	26.98	18.60	25.67	18.55	135.4	18.74	352.2	18.39
	5	124.69	20.96	201.9	20.85	143.78	20.08	1026	19.85	15594	20.05

**Table 8**

The value of the objective function with regard with the corresponding thresholds listed in Table 6.

Image (size)	<i>k</i>	Fitness value				
		MEABCT	MEHBMOT	PSO	HCOCLPSO	Fast Otsu's method
LENA (512 × 512)	2	12.6990	12.6990	12.6990	12.6990	12.6920
	3	15.7658	15.7658	15.7658	15.7658	15.7591
	4	18.5875	18.5875	18.5875	18.5875	18.5084
	5	21.2468	21.2425	21.2400	21.2358	21.1301
PEPPER (512 × 512)	2	12.6348	12.6348	12.6348	12.6348	12.6044
	3	15.6892	15.6892	15.5772	15.6892	15.6829
	4	18.5397	18.5397	18.5395	18.5397	18.4699
	5	21.2830	21.2830	21.2806	21.2804	21.1905
BIRD (256 × 256)	2	11.1647	11.1647	11.1647	11.1647	11.1149
	3	13.8659	13.8659	13.8659	13.8659	13.6677
	4	16.4558	16.4549	16.4543	16.4549	16.1819
	5	18.6961	18.6913	16.4549	18.6520	18.4232
CAMERA (256 × 256)	2	12.1688	12.1688	12.1688	12.1688	11.3367
	3	15.2274	15.2274	15.2274	15.2274	14.3235
	4	18.3995	18.3955	18.3955	18.3955	17.2633
	5	21.0831	21.0691	21.0680	21.0691	19.8241
GOLDHILL (256 × 256)	2	12.5384	12.5384	12.5384	12.5384	12.5343
	3	15.5957	15.5957	15.5957	15.5957	15.5543
	4	18.3957	18.3973	18.3966	18.3931	18.3499
	5	21.0934	21.0817	21.0791	21.0817	20.8941

an idea of complexity, and the PSNR which is used to determine the quality of the thresholded images. Table 5 shows the selected thresholds, computation time, PSNR value and the corresponding fitness value of five test images with different thresholds. This table provides quantitative standard for evaluating. This table shows that the number of thresholds increase, the PSNR and the fitness value are enlarged.

The MEABCT and other four multilevel thresholding methods that are MEHBMOT, PSO, HCOCLPSO and Fast Otsu's algorithm are implemented for the purpose of comparisons. Table 6 shows the selected thresholds of the five test images. It is interesting that the selected thresholds by the MEABCT algorithm are equivalent (for 2- or 3-threshold problems) or very close (4- or 5-threshold problem) to the ones MEHBMOT algorithm; nevertheless, there

are significant differences of selected thresholds with regard to the Fast Otsu's method. This result reveals that the segmentation results depend heavily on the objective function that is selected. Furthermore, the thresholds obtained by PSO algorithms in the segmentation of BIRD image are also distinct from the one of the MEABCT algorithm in 5-level thresholding. It is possible to reveal that the PSO algorithm is unsuitable to search for thresholds. Table 7 shows the computation time and the corresponding PSNR values of the four different multilevel thresholding methods. The fitness values of corresponding thresholds are shown in Table 8. Several aspects are found in the two tables. The computation time of the MEABCT algorithm is between the MEHBMOT and PSO in the segmentation of LENA, PEPPER and BIRD images, but the corresponding PSNRs are greater than the ones of the MEHBMOT and PSO

algorithms. As to the segmentation of CAMERA and GOLDHILL images, the time complexity and PSNR values are superior to other four methods. An aspect is found that the HCOCLPSO and the Fast Otsu's method are not acceptable because of the heavy need of computation times. Finally, from the corresponding the fitness values of selected thresholds using MEABCT algorithm it appears the fact that the selected thresholds of the MEABCT algorithm can effectively find the adequate solutions based on the maximum entropy criterion.

## 5. Conclusion

In this paper, we have proposed a method, called the MEABCT algorithm, for multilevel thresholds selection using the maximum entropy criterion. The MEABCT algorithm simulates the behavior of honey bee foraging to develop the algorithm to select the adequate thresholds for image segmentation. The segmentation results of MEABCT algorithm are promising and it encourages further researches for applying this algorithm to complex and real-time image analysis problems such as automatic target recognition and complex document analysis.

## Acknowledgment

The author would like to thank the National Science council, ROC, under Grant No. NSC 98-2221-E-251-004 for support of this work.

## References

- Abutaleb, A. S. (1989). Automatic thresholding of gray level pictures using two-dimensional entropy. *Computer Vision Graphics Image Process*, 47, 22–32.
- Hammouche, K., Diag, M., & Siarry, P. (2010). A comparative study of various meta-heuristic techniques applied to multilevel thresholding problem. *Engineering Applications of Artificial Intelligence*, 23, 667–688.
- Horng, M. H. (2010). A multilevel image thresholding using the honey bee mating optimization. *Applied Mathematics and Computation*, 215, 3302–3310.
- Kittler, J., & Illingworth, J. (1986). Minimum error thresholding. *Pattern Recognition*, 19, 41–47.
- Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony algorithm. *Applied Soft Computing*, 687–697.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39, 459–471.
- Karaboga, D., & Ozturk, C. (2011). A novel clustering approach: artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 11, 652–657.
- Karaboga, D. (2010). Artificial bee colony algorithm homepage. Access by <<http://mf.erciye.edu.tw/abc/>>.
- Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision Graphics Image Processing*, 29, 273–285.
- Liao, P. S., Chen, T. S., & Chung, P. C. (2001). A fast algorithm for multilevel thresholding. *Journal of Information Science and Engineering*, 17, 713–727.
- Li, G. H., & Lee, C. K. (1993). Minimum cross entropy thresholding. *Pattern Recognition*, 26, 617–625.
- Madhubanti, M., & Amitava, A. (2008). A hybrid cooperative-comprehensive learning based algorithm for image segmentation using multilevel thresholding. *Expert Systems with Application*, 34, 1341–1350.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, Cybernetics*, SMC-9, 62–66.
- Shoo, P. K., Soltani, S., Wong, A. K. C., & Chen, Y. C. (1988). A survey of thresholding techniques. *Computer Vision Graphics and Image Processing*, 41, 233–236.
- Singh, A. (2009). An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing*, 9(2), 625–631.
- Wang, S., Chung, F. L., & Xiong, F. (2008). A novel image thresholding method based on Parzen window estimate. *Pattern Recognition*, 41, 117–129.
- Ye, Z. W., Chen, H., Li, W., & Zhang, J. P. (2008). Automatic threshold selection based on Particle Swarm Optimization algorithm. In *International conference on intelligent computation technology and automation* (pp. 36–39).
- Yin, P. I. (2007). Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Applied Mathematics and Computation*, 184, 503–513.
- Zahara, E., Fan, S.-K. S., & Tsai, D. M. (2005). Optimal multithresholding using a hybrid optimization approach. *Pattern Recognition Letter*, 26, 1082–1095.
- Zhang, R., & Liu, J. (2006). Underwater image segmentation with maximum entropy based on Particle Swarm Optimization (PSO). In *Proceedings of the first international multi-symposiums on computer and computational Sciences (IMSCCS'06)* (pp. 360–363).