

## Research Article

# Artificial Bee Colony Optimizer with Bee-to-Bee Communication and Multipopulation Coevolution for Multilevel Threshold Image Segmentation

Jun-yi Li,<sup>1</sup> Yi-ding Zhao,<sup>2</sup> Jian-hua Li,<sup>1</sup> and Xiao-jun Liu<sup>3</sup>

<sup>1</sup>School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200230, China

<sup>2</sup>School of Computer Science, Zhongyuan University of Technology, Zhengzhou, China

<sup>3</sup>Department of Logistics and Information Management, Zhuhai College of Jilin University, Zhuhai, China

Correspondence should be addressed to Jun-yi Li; [leejy2006@163.com](mailto:leejy2006@163.com)

Received 29 September 2014; Accepted 28 January 2015

Academic Editor: Sishaj P. Simon

Copyright © 2015 Jun-yi Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a modified artificial bee colony optimizer (MABC) by combining bee-to-bee communication pattern and multipopulation cooperative mechanism. In the bee-to-bee communication model, with the enhanced information exchange strategy, individuals can share more information from the elites through the Von Neumann topology. With the multipopulation cooperative mechanism, the hierarchical colony with different topologies can be structured, which can maintain diversity of the whole community. The experimental results on comparing the MABC to several successful EA and SI algorithms on a set of benchmarks demonstrated the advantage of the MABC algorithm. Furthermore, we employed the MABC algorithm to resolve the multilevel image segmentation problem. Experimental results of the new method on a variety of images demonstrated the performance superiority of the proposed algorithm.

## 1. Introduction

Multithreshold image segmentation technique is considered as a simple but effective method to extract useful objects from the digital image background, which required multiple threshold values to accomplish segmentation. The image segmentation can be roughly grouped into optimal threshold [1–4] and property-based threshold [5–7]. The purpose of the first category is to find the proper thresholds for classifying the histogram at a desired level by optimizing an objective function. The second category searches the optimal thresholds by measuring some property of the histogram, which is fast and suitable for the case of multilevel threshold but hard to determine the number of thresholds. Such methods include entropy-based threshold methods, minimization of Bayesian error, and clustering-based methods.

Many existing algorithms utilize the exhaustive search approach to pursue the desired thresholded classes [8–12]. References [8, 9] proposed some novel methods derived from optimizing an objective function. Several algorithms

originally proposed for bilevel threshold have been later extended to multilevel threshold [3, 4, 11, 12]. However, due to the exhaustive search employed, the computational complexity of these methods will rise exponentially when the number of thresholds increases, which limits the multilevel threshold applications.

Recently, owing to their predominant abilities of coping with complex nonlinear optimizations, several swarm intelligence (SI) algorithms as the powerful optimization tools have been applied to address this concerning issue [12–15]. Among them, the artificial bee colony algorithm (ABC) is a popular SI algorithm, which simulates the social foraging behavior of a honeybee swarm [16]. Due to its simple arithmetic and good robustness, the ABC algorithm has been widely used in solving many numerical optimizations and engineering optimization problems [17–20]. However, facing up complex problems, similar to other SIs, ABC algorithm suffers from the following drawbacks [17]. (1) The solution search equation of ABC works well in global exploration but is poor in the exploitation process. (2) With the dimension increasing,

the information exchange of each individual is limited in a random dimension, resulting in a slow convergence rate. (3) Due to the random selection of the neighbor bee and dimensions, food sources with higher fitness are not utilized, which influences the ability of global search.

Thus, this paper presents a modified artificial bee colony algorithm for multilevel threshold image segmentation, namely, MABC, to conquer the original ABC drawbacks to some extent, which synergizes the idea of multipopulation cooperative model with the bee-to-bee communication strategy. In the bee-to-bee communication mechanism, with the enhanced information change strategy, individuals can share more information from the elites in the early exploration stage of the algorithm. With the multipopulation cooperative approach, the hierarchical colony with different topologies can be structured, which can maintain diversity of the whole community. By incorporating this hybrid strategy, the proposed MABC can be claimed very powerful, because the exploitation and exploration can be elaborately balanced, which guarantees finding the optimal thresholds at a more reasonable time.

The rest of the paper is organized as follows. Section 2 describes the canonical ABC algorithm in detail. In Section 3, the proposed modified artificial bee colony (MABC) algorithm is given. Section 4 presents the experimental studies of the proposed MABC and the other algorithms with descriptions of the involved benchmark functions, experimental settings, and experimental results. And its application to image segmentation has been presented in Section 5. Finally, Section 6 outlines the conclusion.

## 2. Canonical ABC Algorithm

The artificial bee colony (ABC) algorithm was proposed by Karaboga in 2005 [16], simulating the intelligent foraging behavior of a honeybee swarm. The entire bee colony contains three groups of bees: employed bees, onlookers, and scouts [17, 18]. Employed bees are responsible for exploring the specific food sources; meanwhile they pass the food information to onlooker bees. The number of employed bees is equal to that of food sources. Then onlooker bees choose good food sources based on the received information, and then they further exploit the food near their selected food source. The food source with higher quality would have a larger opportunity to be selected by onlookers. There is a control parameter called “*limit*” in the canonical ABC algorithm. If a food source is not improved anymore when *limit* is exceeded, it is assumed to be abandoned by its employed bee and the employed bee associated with that food source becomes a scout to search for a new food source randomly. The fundamental mathematic representations are listed as follows.

*Step 1* (initialization phase). In initialization phase, a group of food sources are generated randomly in the search space using the following equation:

$$x_{i,j} = x_j^{\min} + \text{rand}(0, 1) (x_j^{\max} - x_j^{\min}), \quad (1)$$

where  $i = 1, 2, \dots, SN$ ,  $j = 1, 2, \dots, D$ .  $SN$  is the number of food sources.  $D$  is the number of variables, that is, problem dimension.  $x_j^{\min}$  and  $x_j^{\max}$  are the lower upper and upper bounds of the  $j$ th variable, respectively.

*Step 2* (employed bees' phase). In the employed bees' phase, the neighbor food source (candidate solution) can be generated from the old food source of each employed bee in its memory using the following expression:

$$v_{i,j} = x_{i,j} + \phi (x_{i,j} - x_{k,j}), \quad (2)$$

where  $k$  is a randomly selected food source and must be different from  $i$ ;  $j$  is a randomly chosen index;  $\phi$  is a random number in range  $[-1, 1]$ .

*Step 3* (onlooker bees' phase). In the onlooker bees' phase, an onlooker bee selects a food source depending on the probability value associated with that food source;  $P_i$  can be calculated as follows:

$$P_i = \frac{\text{fitness}_i}{\sum_{j=1}^{SN} \text{fitness}_j}, \quad (3)$$

where  $\text{fitness}_i$  is the fitness value of the  $j$ th solution.

*Step 4* (scout bees' phase). In scout bees' phase, if a food source cannot be improved further through a predetermined cycle (called “*limit*” in ABC), the food source is supposed to be abandoned. The employed bee subsequently becomes a scout. A new food source will be produced randomly in the search space using (1).

The employed, onlooker, and scout bees' phase will recycle until the termination condition is met.

## 3. Modified Artificial Bee Colony Algorithm

### 3.1. Bee-to-Bee Communication Model

*3.1.1. Von Neumann Topology Structure.* The inherent property of swarm intelligence is the “population” where unsophisticated individuals share global information with their neighborhoods to create collective intelligence in some specific topologies [21, 22]. The spatial topology structure significantly affects the spreading path of information flow across the structured population and also determines the social interaction between individuals.

Recently, using population topology technique to improve the performances of evolutionary algorithms has been strongly suggested [23–25]. In [23], after deeply investigating the relationship between various topologies and performances, Kennedy drew an interesting conclusion that Von Neumann topology demonstrates superior performances on a wide range of testing functions, as shown in Figure 1(a).

In ABC, from (1), it is clearly visible that any candidate neighborhood of individual  $i$  can be selected randomly from the entire population, which means every individual is connected to individual  $i$ . In other words, the population

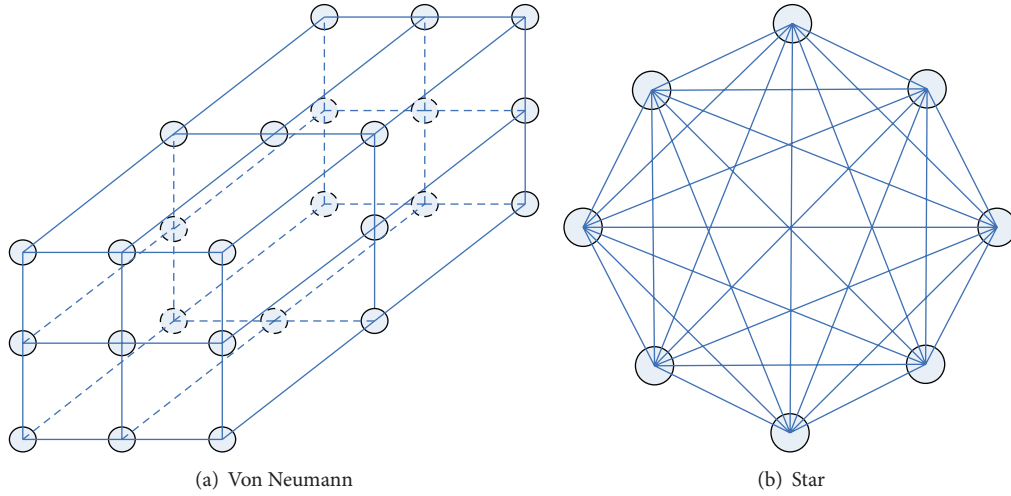


FIGURE 1: Two classical population topology structures.

topology of ABC belongs to the Star type, as shown in Figure 1(b). As depicted in [23], it was theorized that although the connectivity of the Von Neumann structure is lower than the Star type, the Von Neumann type can enable individuals of population to cover a larger search space and to maintain diversity of population, avoiding falling into the deep local optimum. The detailed steps of constructing the Von Neumann structure are given as shown in Algorithm 1.

In the MABC algorithm, the Von Neumann model is utilized in the employed bee phase where each individual  $i$  has three neighborhoods, namely, the above node, the left node, and the below node. And in the onlooker bee phase and the scout bee phase, the original topology is still employed. This can maintain an appropriate balance between improving convergence rate and enhancing diversity of population.

**3.1.2. Enhanced Information Exchange Based on Crossover.** In canonical ABC, from (2), we can see that the information exchange between individuals is restricted to a narrow scope, which relies on a single random dimension of a single random bee. To address this, the crossover-based information sharing technique is employed in MABC. To benefit from the global best information derived from the entire population, a given number of elites from excellent individuals are subjected to crossover operation. The underlying idea behind this method is to facilitate better search in complex higher dimensional search space as opposed to classical ABC that perturbs a single parameter. The steps for crossover operation phase are given as follows and the schematic diagram is illustrated in Figure 2.

**Algorithm 2** (crossover operation). Consider the following.

**Step 1** (select elites to the best-performing list (BPL)). A set of individuals with higher fitness from current food sources are chosen to construct the best-performing list (BPL), whose size is equal to the current population size.

**Step 2** (crossover operation). Parents are chosen from the BPL's elites only for crossover by the tournament selection scheme. Then, two offspring are generated by the arithmetic crossover operation:

$$P_{\text{new}} = r\text{Parent 1} + (1 - r)\text{Parent 2}, \quad (4)$$

where  $P_{\text{new}}$  is new offspring, Parent 1 and Parent 2 are randomly chosen from BPL, and  $r$  is a random coefficient in range of  $[0, 1]$ .

**Step 3** (update population). A selecting rate CR is used to determine the replaced individuals in the population. If population size is  $S$ , then the replaced individuals size is  $S * \text{CR}$ , and they are chosen randomly from the population. Through the greedy selection between the replaced individual  $P_j$  and the new offspring  $P_{\text{new}}$ , the better one remains in the population.

**3.2. Multipopulation Cooperative Mechanism.** The multipopulation cooperative concept has been widely adopted to improve the search ability of algorithms via decomposing complex problems into simple tasks tackled in parallel. With the multipopulation cooperative mechanism as shown in Figure 3, the single population in ABC can be structured into two levels with different topologies and information exchange strategies. The new characteristics can be abstracted as follows.

In each foraging process, this two-level multipopulation evolution can be formulated as follows:

$$x_{\text{new},i,j} = x_{i,j} + \varphi_1 (x_{\text{ibest},j} - x_{i,j}) + \varphi_2 (x_{\text{pbest},j} - x_{i,j}), \quad (5)$$

where  $x_{\text{ibest}}$  is the global best individual within species and  $x_{\text{sbest}}$  is the best individual found so far among all species. From (5), we can see that the term  $x_{\text{ibest}}$  represents the information exchange within species (individual level) in a specific topology, and  $x_{\text{pbest}}$  denotes the cooperation of agents

Step 1. Split the population of  $SN$  individuals into  $M$  rows and  $N$  cols, and  $SN = M * N$ .  
 Step 2. For  $i = 1 : SN$ .  
 Step 3.  $N_i(1) = (i - N) \bmod SN$   
           if  $N_i(1) == 0$   
            $N_i(1) = SN$ .  
 Step 4.  $N_i(2) = i - 1$ ,  
           if  $(i - 1) \bmod N == 0$ ,  
            $N_i(2) = i - 1 + N$ .  
 Step 5.  $N_i(3) = i + 1$ ,  
           if  $i \bmod N == 0$ ,  
            $N_i(3) = i + 1 - N$ .  
 Step 6.  $N_i(4) = (i + N) \bmod SN$ ,  
           if  $N_i(4) == 0$ ,  
            $N_i(4) = SN$ .  
 Step 7.  $nhood(i) = [N_i(1), N_i(2), N_i(3), N_i(4)]$ .  
       End For

ALGORITHM 1: Construction method of Von Neumann.

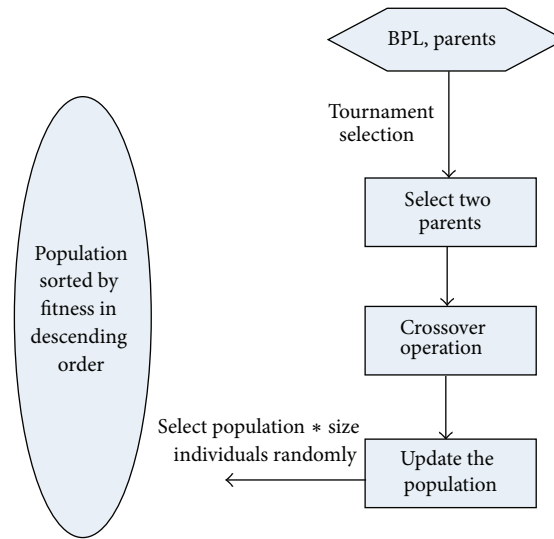


FIGURE 2: Crossover-based information sharing strategy.

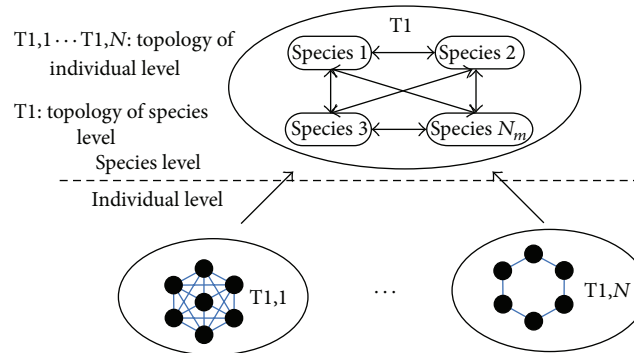


FIGURE 3: Multispecies coevolution mechanism.

between species (species level). This will help to amplify the diversity of population. Furthermore, the information exchange strategy significantly improves the search ability of the proposed method.

**3.3. The Proposed Algorithm.** By incorporating the aforementioned evolution model and strategies, namely, multipopulation cooperative model and bee-to-bee communication strategy, each individual can regulate its trajectory according to the effective topology and information exchange strategy. In particular, the population evolution process can make use of the historical experience of individual level and the global best information of the species level, which can maintain the diversity of population. The main steps of the proposed algorithm are given as the following processes.

*Algorithm 3* (the proposed MABC algorithm). Consider the following.

*Step 1* (initialization). Consider the following.

*Step 1.1.* Randomly generate  $N$  species, each possessing  $SN$  individuals.

*Step 1.2.* Initialize the population topology structure according to Algorithm 1.

*Step 1.3.* Evaluate the fitness of each bee.

*Step 1.4.* Set maximum cycles ( $LimitC$ ).

*Step 2.* Consider iteration = 0.

*Step 3* (employ bee phase). Loop over each food source.

*Step 3.1.* Generate a candidate solution  $V_i$  by (4) and evaluate  $f(V_i)$ .

*Step 3.2.* Greedy selection and memorize the better solution.

*Step 4.* Calculate the probability value  $p_i$  by (3).

*Step 5* (onlooker bee phase). Consider the following.

*Step 5.1.* Select a food source by  $p_i$  and generate a candidate solution  $V_i$  by (4).

*Step 5.2.* Greedy selection and memorize the better solution.

*Step 6* (crossover phase). Consider the following.

*Step 6.1.* Select elites to the best-performing list (BPL) as Algorithm 2.

*Step 6.2.* Crossover and mutation operation as Algorithm 2.

*Step 6.3.* Update with greedy selection strategy as Algorithm 2.

*Step 7.* Consider iteration = iteration + 1.

*Step 8.* If the iteration is greater than  $LimitC$ , stop the procedure; otherwise, go to Step 3.

*Step 9.* Output the best solution achieved.

TABLE 1: Parameters of the MABC.

MABC = (S, D, N, CR)	
S	Population size
D	Dimension of problem
N	Species number
CR	Selection rate

In summary, in order to facilitate the below presentation and test formulation, we define unified parameters for MABC model in Table 1 and the flowchart of MABC algorithm is summarized in Figure 4.

## 4. Benchmark Test

In the experimental studies, 15 benchmark functions are employed to test the proposed MABC algorithm [26–30]. The involved benchmarks can be classified as basic benchmarks ( $f_1$ – $f_8$ ) and CEC 2005 benchmarks ( $f_9$ – $f_{15}$ ). The formula for each basic benchmark and CEC 2005 test functions is shown in Tables 2 and 3. In our test, the function evaluations (FEs) criterion is used as a time measure substituting the number of iterations, due to the fact that the algorithms do differing amounts of work in their inner loops.

**4.1. Parameters Settings for the Involved Algorithms.** Experiment is conducted to compare with successful EA and SI algorithms, including original artificial bee colony algorithm (ABC) [17], canonical PSO with constriction factor (PSO) [31], genetic algorithm with elitism (EGA) [32], and covariance matrix adaptation evolution strategy (CMA-ES) [33]. The independent run time is 30, and the maximum evaluation number is 100000. For the involved benchmarks, the dimensions are all set as 30.

All related parameters of the involved compared algorithms are set as their original literatures suggest: initialization conditions of CMA-ES are referred to in [33], and the candidate offspring number per time step is set as  $\lambda = 4\mu$ . For ABC, the limit parameter is  $SN \times D$ , where  $SN$  is employed bee number and  $D$  is the benchmark dimension; the learning rates  $c_1$  and  $c_2$  of PSO are both 2.05 and the constriction factor  $\chi = 0.729$ ; the crossover rate and mutation rate of EGA are 0.8 and 0.01, respectively, and the global elite with a rate of 0.06 is used [31]. The parameter setting for all algorithms is listed in Table 4. For the proposed MABC, the species number  $N$  and selection rate  $CR$  should be tuned firstly in the next section.

### 4.2. Sensitivity in Relation to Parameters of MABC

**(1) Effect of Agent Number  $N$  in Species Level.** In order to study the sensitivity in relation to the parameter  $N$ , six classical 30-dimensional benchmark functions were employed, namely, unimodal sphere ( $f_1$ ), unimodal nonseparable Rosenbrock ( $f_2$ ), multimodal Schwefel ( $f_3$ ), unimodal shifted sphere ( $f_9$ ), multimodal rotated Rastrigin ( $f_{12}$ ), and multimodal nonseparable rotated Ackley ( $f_{13}$ ) functions, as defined in Table 2.



TABLE 2: Classical test suite.

Name	Function	Limits
Sphere ( $f_1$ )	$f_1(x) = \sum_{i=1}^D x_i^2$	$x_i \in [-5.12, 5.12]^D$
Rosenbrock ( $f_2$ )	$f_2(x) = \sum_{i=1}^D 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2$	$x_i \in [-2.048, 2.048]^D$
Quadric ( $f_3$ )	$f_3(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	$x_i \in [-65.536, 65.536]^D$
Sin ( $f_4$ )	$f_4(x) = \frac{\pi}{n} \left\{ 10 \sin^2 \pi x_1 + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + 10 \sin^2 \pi x_{i+1}) + (x_n - 1)^2 \right\}$	$x_i \in [-10, 10]^D$
Rastrigin ( $f_5$ )	$f_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) + 10$	$x_i \in [-5.12, 5.12]^D$
Schwefel ( $f_6$ )	$f_6(x) = D * 418.9829 + \sum_{i=1}^D -x_i \sin(\sqrt{ x_i })$	$x_i \in [-500, 500]^D$
Ackley ( $f_7$ )	$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$	$x_i \in [-32, 32]^D$
Griewank ( $f_8$ )	$f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$x_i \in [-5.12, 5.12]^D$

TABLE 3: CEC 2005 test suite.

Name	Function	Limits
Shifted Sphere ( $f_9$ )	$f_9(x) = \sum_{i=1}^D z_i^2 + f_{\text{bias1}}, \quad z = x - o$	$x_i \in [-100, 100]^D$
Shifted Schwefel's Problem 1.2 ( $f_{10}$ )	$f_{10}(x) = \sum_{i=1}^D \left( \sum_{j=1}^i z_j \right)^2 + f_{\text{bias}}, \quad z = x - o$	$x_i \in [-100, 100]^D$
Shifted Rosenbrock's function ( $f_{11}$ )	$f_{11}(x) = \sum_{i=1}^{D-1} \left( 100(z_i^2 - z_{i+1})^2 + (z_i^2 - 1)^2 \right) + f_{\text{bias}}$	$x_i \in [-100, 100]^D$
Shifted rotated Griewank without bounds ( $f_{12}$ )	$f_{12}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}$	No bounds
Shifted rotated Ackley ( $f_{13}$ )	$f_{13}(x) = -\exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e + f_{\text{bias}}, \quad z = (x - o) * M$	$x_i \in [-32, 32]^D$
Shifted Rastrigin ( $f_{14}$ )	$f_{14}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}}, \quad z = x - o$	$x_i \in [-5, 5]^D$
Shifted rotated Rastrigin ( $f_{15}$ )	$f_{15}(x) = (1 + 0.1 \times \text{rand}[-1, 1]) \sum_{i=1}^D x_i^2$	$x_i \in [-5, 5]^D$

TABLE 4: Parameters setting for all algorithms.

Type	MABC	ABC	PSO	EGA	CMA-ES
S	5	50	50	50	50
N	10	NA	NA	NA	NA
CR	1	NA	NA	NA	NA
$\chi$	NA	NA	0.729	NA	NA
c1	NA	NA	2	NA	NA
c2	NA	NA	2	NA	NA
$\mu/c_\sigma/d_\sigma/c_\epsilon/c_{\text{cov}}$	NA	NA	NA	NA	12/0.1/20/0.12/0.08
Crossover/mutation rate	NA	NA	NA	0.8/0.01	NA

TABLE 5: Results of MABC on benchmarks  $f_1$ – $f_5$  with different CR. In bold are the best results.

Func.		0.05	0.1	0.2	0.4	0.6	1
$f_1$	Mean	<b>0</b>	2.8168E – 220	3.8569E – 290	5.7424e – 652	3.7243E – 490	2.3124E – 900
	Std.	<b>0</b>	4.5659E – 220	6.0881E – 290	7.6232e – 651	3.0343E – 483	1.3126E – 900
$f_2$	Mean	1.9134E – 03	1.1004E – 06	5.9872E – 07	4.8661E – 01	5.5272E – 07	<b>3.5209E – 10</b>
	Std.	1.9216E – 03	5.2065E – 06	2.8703E – 06	<b>2.7358E – 05</b>	4.1818E – 06	8.0095E – 05
$f_3$	Mean	2.0120E – 32	1.8149E – 34	3.4981E – 123	1.5024E – 65	2.3232E – 89	<b>4.6971E – 200</b>
	Std.	1.1004E – 32	1.0265E – 34	6.1778E – 123	1.5024E – 65	1.4545E – 90	<b>1.3015E – 200</b>
$f_4$	Mean	1.1004E – 12	6.2083E – 53	1.5024E – 32	1.7491E – 43	1.49091E – 32	<b>1.5183E – 65</b>
	Std.	4.8123E – 12	<b>7.6701E – 65</b>	1.5024E – 32	8.7679E – 43	1.51515E – 32	6.9492E – 65
$f_5$	Mean	1.9216E – 11	4.6398E – 19	2.6236E – 14	3.8570E – 21	4.49697E – 45	<b>1.0185E – 50</b>
	Std.	1.0183E – 11	3.6544E – 19	2.7469E – 14	2.7245E – 21	1.72121E – 46	<b>1.1018E – 50</b>

In this test, the values of parameter  $N$  were changed from 1 to 15, while the selection rate CR is preset as 1. Results of this experiment are illustrated in Figure 5. From Figure 5, it is clearly visible that when  $N$  is near 10, MABC performed superior in terms of mean values and convergence rate on most test functions. However, the effect of this parameter is not very obvious in the test.

(2) *Effect of Selection Rate CR.* Five basic benchmark functions ( $f_1$ – $f_5$ ) are employed to evaluate the performance of MABC variants with different CRs. All the functions are implemented for 30 run times, whose dimension is 30. From Table 5, we can observe that MABC variant with CR equal to 1 performed best on four functions among all five functions while the variant with CR equal to 0.05 obtained best results on function  $f_4$ . Therefore, the value of selection rate CR in each crossover operation can be set to 1 as an optimal value in all the following experiments.

#### 4.3. Numerical Results and Comparison

4.3.1. *Results for the Classical Benchmarks.* The means and standard deviations of the 30 run times of the nine involved algorithms on the classical test suite are listed in Table 6. The best results among those algorithms are shown in bold in Table 6. Figures 6(a)–6(g) present the average convergence rates of each algorithm for each benchmark.

From Table 6, MABC and ABC obtained satisfactory results on the unimodal functions ( $f_1$  to  $f_3$ ) in terms of accuracy and convergence. MABC performs a little worse than ABC on these functions but significantly better than other algorithms.  $f_5$ – $f_8$  are the most commonly used test multimodal functions, and an algorithm can be easily trapped in a local minimum. As expected, the MABC gets more favorable results than the compared algorithms on all these cases (shown in Table 6 and Figure 6). The differences between the compared algorithms on these multimodal functions suggest that the MABC is better at a fine-gained search than the other algorithms. The key difference is mainly due to the specific topology technique and crossover operator in the MABC. The crossover operator helps to enlarge the information exchange space while accelerating the searching process. The multipopulation strategy makes full use of the global

information and transfers this information effectively across the population structured by the Von Neumann topology. From the overall results on the eight classical functions, MABC achieved better results on most functions than its counterpart ABC and other algorithms due to its usage of the proposed strategies.

In order to further investigate the efficacy and robustness of the MABC, the analysis of variance (ANOVA) test was employed to determine the statistical characteristics of each proposed algorithm over the others. The box plots shown in Figures 7(a)–7(h) demonstrate the statistical performance of each involved algorithm on the classical test suite for 30 runs individually. According to the red lines in Figures 7(a)–7(h), which refer to the low median values obtained by each algorithm, it is clearly visible and proven that MABC achieved good variance distribution of compromise solutions on all classical functions. Note that the ABC algorithm also obtained satisfactory results on some classical functions.

4.3.2. *Results for the CEC 2005 Benchmarks.* Seven shifted and rotated functions  $f_9 \sim f_{15}$  from CEC 2005 test bed are viewed to be much more difficult to resolve than basic benchmarks. The experimental results in terms of means, standard deviations, and ranks are presented in Table 7. From these results, it can be observed that MABC performed best on five out of the seven functions. ABC and CMA-ES achieved similar ranking, only worse than MABC. Furthermore, MABC can obtain the global optimum for  $f_{11}$  and  $f_{14}$  within 10000 FEs. It is clearly visible and proven that MABC performed more powerful on CEC 2005 test suite compared to its performance on basic benchmarks, which means that MABC using three effective strategies is more competent in tackling complex problems.

The superior performance by MABC on CEC 2005 benchmarks can be explained as follows: when other involved algorithms are trapped in local optima, the MABC uses the enhanced information change strategy successfully to jump out of the deep local optimum. By hybridizing the multipopulation cooperative model with the bee-to-bee communication mechanism, the individuals of MABC can enlarge the search space and maintain proper diversity of the population. This individual search behavior is guided by the global best experience from different species.

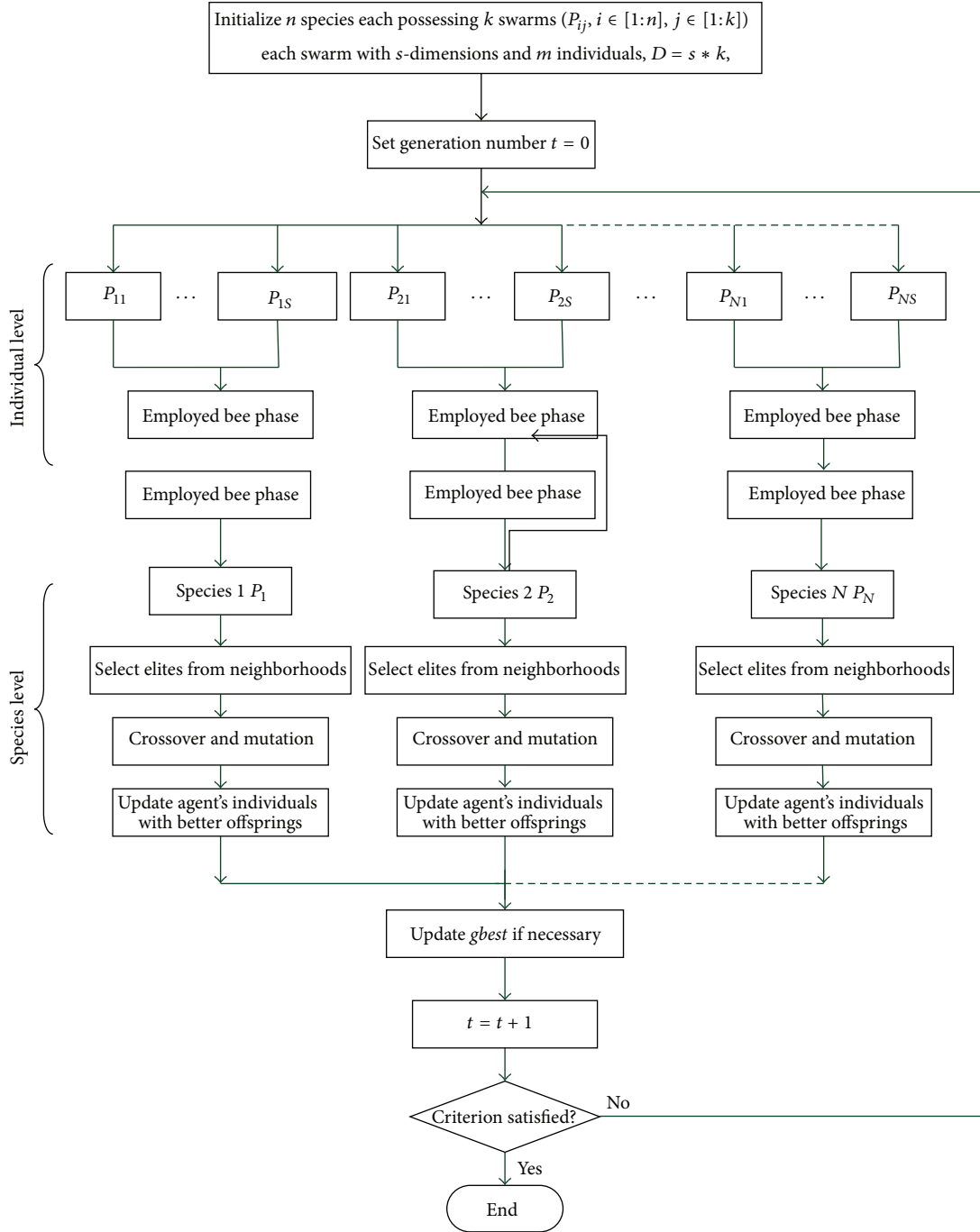


FIGURE 4: Flowchart of the MABC algorithm.

## 5. Multilevel Threshold for Image Segmentation by MABC

**5.1. Entropy Criterion-Based Fitness Measure.** The Otsu multithreshold entropy measure [34] proposed by Otsu has been popularly employed in determining whether the optimal threshold method can provide image segmentation with satisfactory results. Here, it is used as the objective function for the involved algorithms and its process can be described as follows.

Let the gray levels of a given image range over  $[0, L - 1]$  and  $h(i)$  denote the occurrence of gray level  $i$ .

Let

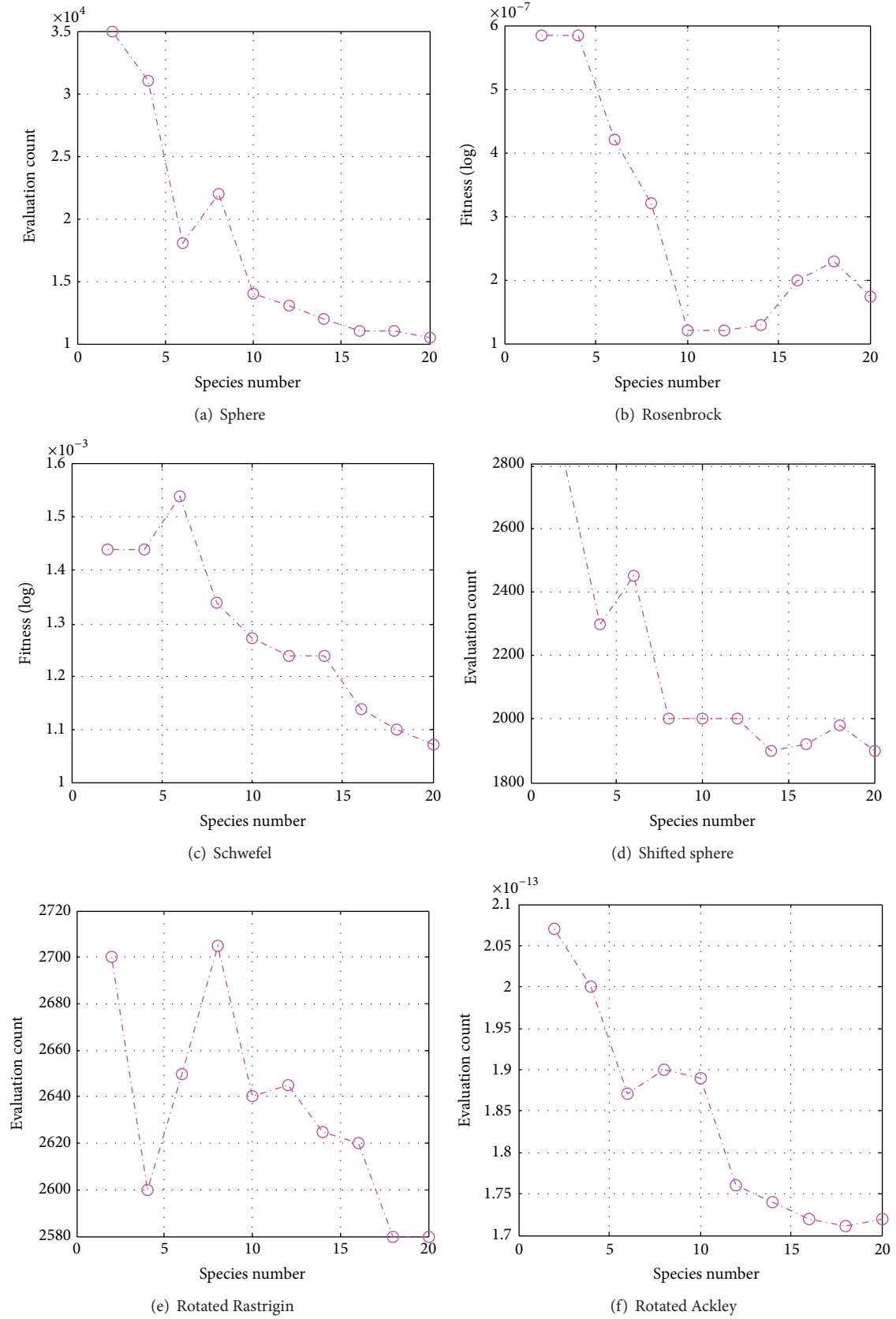
$$N = \sum_{i=0}^{L-1} h(i),$$

$$P(i) = \frac{h(i)}{N} \quad (6)$$

$$\text{for } 0 \leq i \leq L - 1$$

$$\text{Maximize } f(t) = w_0 w_1 (u_0 - u_1)^2, \quad (7)$$



FIGURE 5: MABC's results on six test functions with different  $N$ .

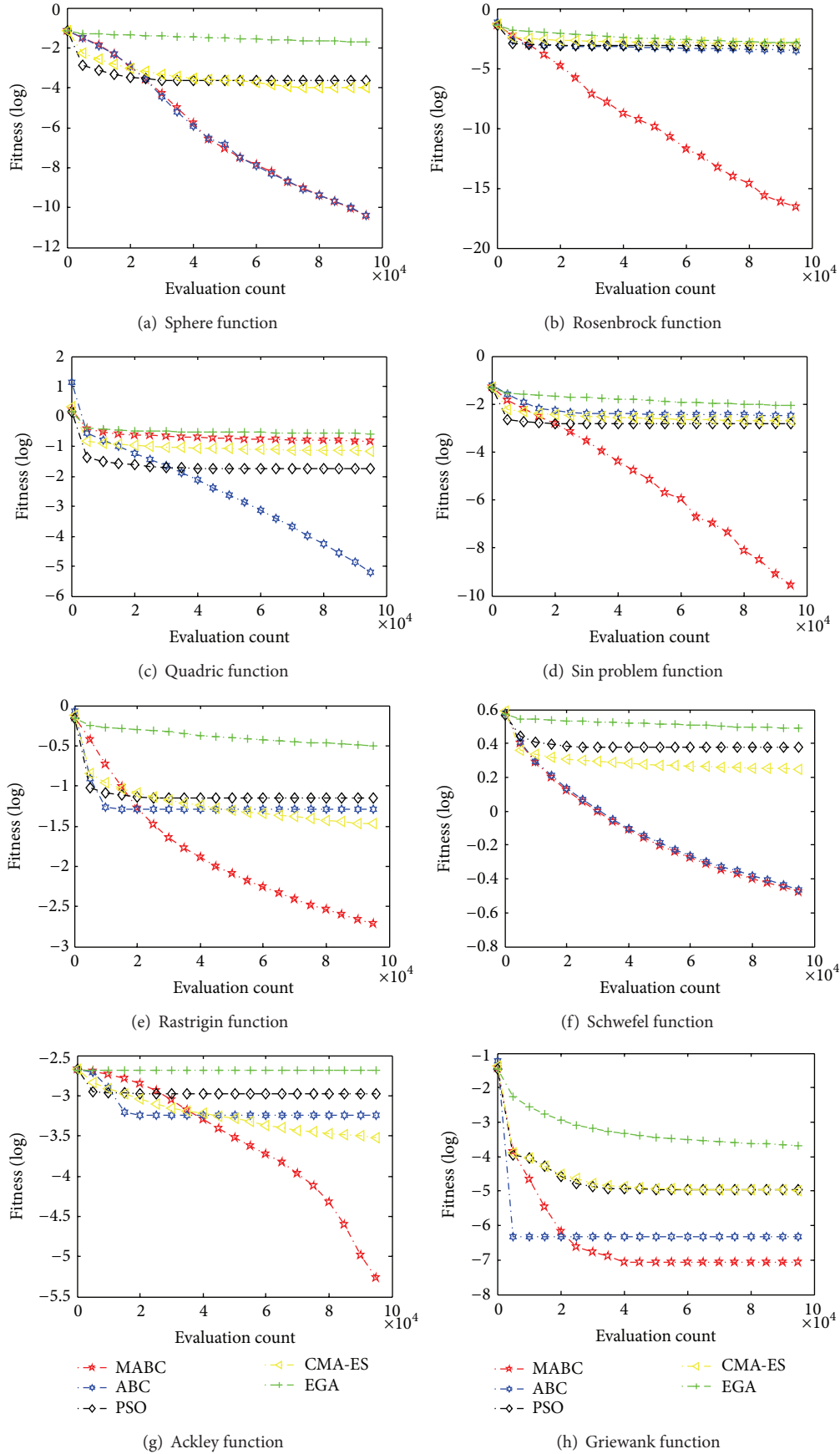


FIGURE 6: The median convergence results of classical test functions.

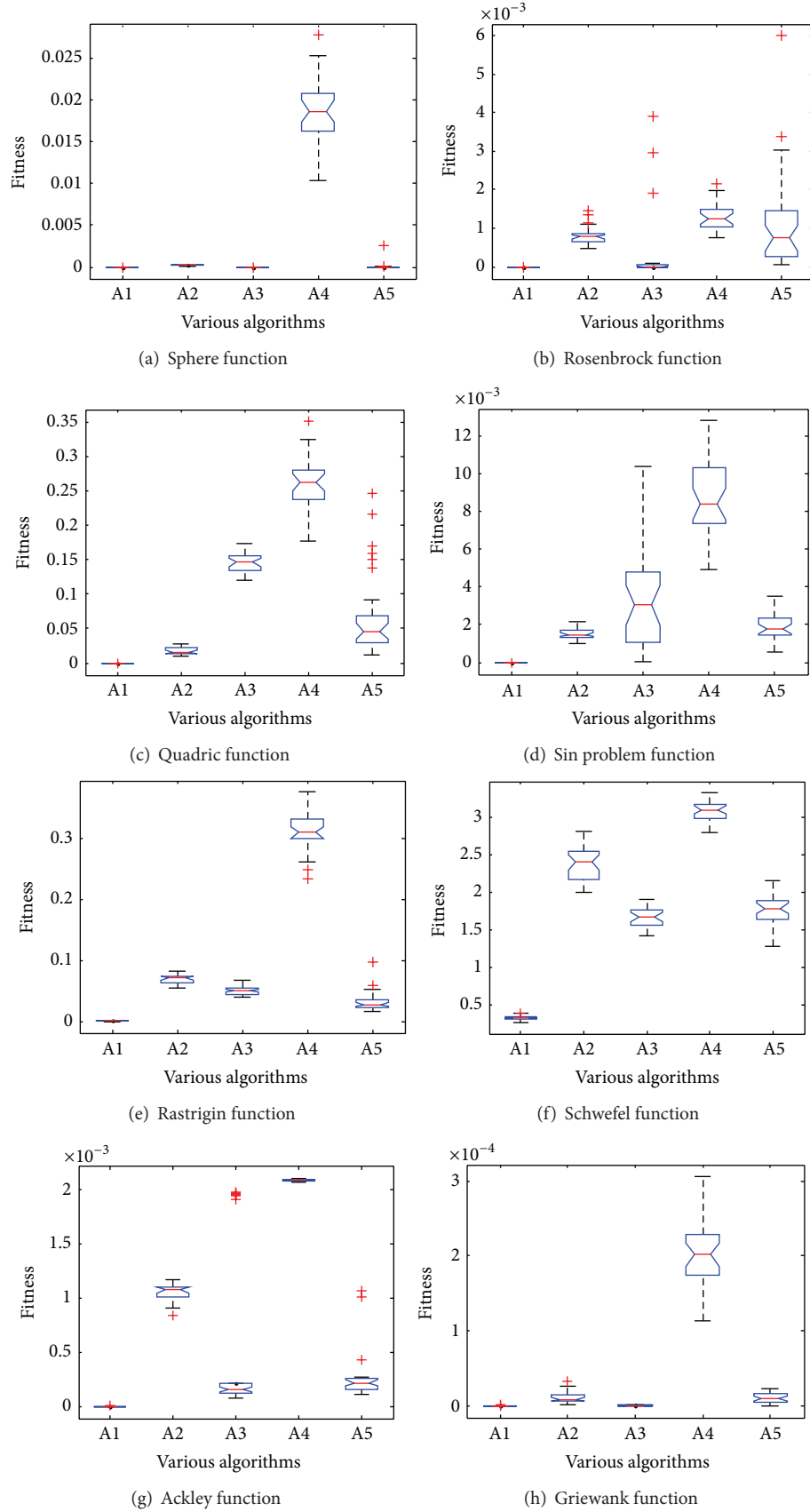


FIGURE 7: ANOVA results of all algorithms. Here 1 to 5 is index of MABC, ABC, PSO, CMA-ES, and EGA, respectively.

TABLE 6: Performance of all algorithms on classical test suite. In bold are the best results.

Functions		MABC	ABC	PSO	CMA-ES	GA
$f_1$	Mean	$1.7351E-11$	<b><math>1.6434E-011</math></b>	$2.3212E-004$	$1.8859E-06$	$1.4189E-03$
	Std.	$1.9635E-11$	<b><math>1.2344E-011</math></b>	$4.7645E-004$	$4.2722E-04$	$1.9446E-03$
	Rank	2	<b>1</b>	4	3	5
$f_2$	Mean	<b><math>5.7820E-15</math></b>	$1.4323E-004$	$2.2123E-004$	$6.3078E-03$	$5.3922E-02$
	Std.	<b><math>1.2471E-10</math></b>	$2.3231E-014$	$1.1231E-005$	$2.2137E-06$	$5.1350E+01$
	Rank	<b>1</b>	2	3	4	5
$f_3$	Mean	$1.4690E-01$	<b><math>4.4323E-006</math></b>	$1.8023E-002$	$1.4737E-01$	$2.6220E-01$
	Std.	$1.4675E-02$	<b><math>1.9323E-006</math></b>	$5.2123E-003$	$1.4698E-02$	$3.9397E-02$
	Rank	4	<b>1</b>	2	3	5
$f_4$	Mean	<b><math>1.3181E-10</math></b>	$6.4523E-003$	$1.5012E-003$	$3.3633E-03$	$3.5352E-02$
	Std.	<b><math>1.6126E-10</math></b>	$2.5323E-002$	$2.9146E-004$	$2.6175E-03$	$2.1175E-03$
	Rank	<b>1</b>	4	2	3	5
$f_5$	Mean	$1.7678E-03$	$6.7654E-002$	$8.9773E-002$	<b><math>4.2015E-04</math></b>	$3.1313E-01$
	Std.	$5.3777E-04$	$5.04343E-004$	$7.3045E-003$	<b><math>7.1350E-05</math></b>	$2.9879E-02$
	Rank	2	3	4	<b>1</b>	5
$f_6$	Mean	<b><math>3.2762E-05</math></b>	$3.3212E-005$	$1.7012$	$4.4267E-02$	$3.1018E+00$
	Std.	<b><math>3.1623E-02</math></b>	$3.2121E-002$	$1.4123E-001$	$2.3717E-02$	$1.3784E-01$
	Rank	<b>1</b>	2	4	3	5
$f_7$	Mean	<b><math>3.7026E-06</math></b>	$3.1232E-003$	$1.1212E-002$	$1.7013E-03$	$2.1081E-02$
	Std.	<b><math>2.8588E-06</math></b>	$2.5975E-006$	$7.2444E-005$	$7.8365E-04$	$8.7781E-06$
	Rank	<b>1</b>	3	4	2	5
$f_8$	Mean	<b><math>4.5062E-10</math></b>	$8.4321E-008$	$8.5321E-005$	$2.1953E-07$	$5.3993E-04$
	Std.	<b><math>2.0212E-10</math></b>	$3.3412E-007$	$1.1242E-005$	$4.4017E-07$	$1.9397E-04$
	Rank	<b>1</b>	2	4	3	5

TABLE 7: Performance of all algorithms on CEC05 test suite. In bold are the best results.

Functions		MABC	ABC	PSO	CMA-ES	EGA
$f_9$	Mean	<b><math>-4.4952E+02</math></b>	$-4.3032E+02$	$3.1235E+01$	$-4.2120E+02$	$-3.4432E+02$
	Std.	<b><math>1.8526E-14</math></b>	$2.134E-14$	$5.6723E+02$	$5.1026E-14$	$1.8797E+01$
	Rank	<b>1</b>	2	5	3	4
$f_{10}$	Mean	$-4.4923E+01$	$-4.2743E+01$	$9.3412E+02$	<b><math>-4.500E+02</math></b>	$1.3232E+04$
	Std.	$2.0348E+02$	$2.2354E+02$	$2.78385E+03$	<b><math>2.1216E-14</math></b>	$5.4231E+03$
	Rank	2	3	4	<b>1</b>	5
$f_{11}$	Mean	<b><math>3.90E+02</math></b>	$4.0154E+02$	$2.3541E+07$	$4.1210E+02$	$3.7887E+04$
	Std.	<b><math>2.1158E+00</math></b>	$1.8765E+00$	$2.5536E+07$	$1.7217E+00$	$3.2344E+04$
	Rank	<b>1</b>	2	5	3	4
$f_{12}$	Mean	<b><math>-1.7934E+02</math></b>	$2.2054E+03$	$6.4455E+03$	$2.3212E+03$	$3.1218E+03$
	Std.	<b><math>7.2077E-03</math></b>	$1.3354E-02$	$2.1218E+02$	$1.2323E-03$	$4.2356E+02$
	Rank	<b>1</b>	3	6	2	5
$f_{13}$	Mean	$-1.1866E+02$	$-1.7212E+02$	$-1.7432E+02$	<b><math>-1.1798E+02</math></b>	$-1.7076E+02$
	Std.	$1.3588E-01$	$6.4365E-02$	$1.4523E-01$	<b><math>3.8043E-02</math></b>	$2.4323E-02$
	Rank	2	3	4	<b>1</b>	5
$f_{14}$	Mean	<b><math>-3.300E+02</math></b>	$-3.3000E+02$	$-2.6021E+02$	$-2.6012E+02$	$-2.5087E+02$
	Std.	<b><math>1.2490E-14</math></b>	$3.7352E-14$	$3.4121E+01$	$2.1215E+01$	$3.4341E+00$
	Rank	<b>1</b>	1	4	3	5
$f_{15}$	Mean	<b><math>-2.8580E+02</math></b>	$-1.2853E+02$	$-1.9012E+02$	$-1.6562E+02$	$-1.4098E+02$
	Std.	<b><math>1.1493E+01</math></b>	$3.5421E+01$	$2.4212E+01$	$2.7032E+01$	$4.0121E+01$
	Rank	<b>1</b>	5	2	3	4

TABLE 8: Results obtained by the Otsu method.

Image	$M - 1 = 2$		$M - 1 = 3$		$M - 1 = 4$	
	Objective values	Optimal thresholds	Objective values	Optimal thresholds	Objective values	Optimal thresholds
Avion	$3.4795E + 04$	112,171	$3.4907E + 04$	92,141,189	$3.4954E + 04$	82,125,169,201
House	$2.8810E + 04$	103,172	$2.8862E + 04$	86,138,179	$2.8912E + 04$	769,116,151,184
Lena	$1.7792E + 04$	92,149	$1.7979E + 04$	82,125,169	$1.8043E + 04$	72,111,143,178
Hunter	$9.3798E + 03$	50,113	$9.5403E + 03$	33,82,131	$9.5971E + 03$	33,70,109,144
Mean CPU time	1.5325		$6.1265E + 01$		$2.6553E + 03$	

TABLE 9: The mean CPU time of the compared population-based methods on Otsu algorithm.

Dim.	Alg.		PSO	CMA-ES	EGA
	MABC	ABC			
2 Dim.	$2.3530E - 01$	$9.0000E - 01$	$9.8404E - 01$	$2.8139E - 01$	$2.7958E + 00$
3 Dim.	$2.7218E - 01$	$9.4418E - 01$	$1.0032E + 00$	$3.1475E - 01$	$3.6021E + 00$
4 Dim.	$2.6716E - 01$	$9.8769E - 01$	$1.0086E + 00$	$3.4569E - 01$	$4.9494E + 00$
5 Dim.	$3.0271E - 01$	$1.0054E + 00$	$7.2100E - 01$	$3.5227E - 01$	$1.2512E + 00$
7 Dim.	$3.2453E - 01$	$1.0978E + 00$	$1.1213E + 00$	$4.9490E - 01$	$1.8731E + 00$
9 Dim.	$4.8605E - 01$	$1.0440E + 00$	$1.1448E + 00$	$4.9637E - 01$	$2.6206E + 00$

where

$$\begin{aligned}
 w_0 &= \sum_{i=0}^{t-1} P_i; \\
 u_0 &= \sum_{i=0}^{t-1} i \times \frac{P_i}{w_0}, \\
 w_1 &= \sum_{i=t}^{L-1} P_i; \\
 u_1 &= \sum_{i=t}^{L-1} i \times \frac{P_i}{w_1}
 \end{aligned} \tag{8}$$

and the optimal threshold is the gray level that maximizes (7). Then (7) can also be written as

$$f(t) = \delta^2 - w_0 \times \delta_0^2 - w_1 \times \delta_1^2, \tag{9}$$

where  $w_0$ ,  $w_1$ ,  $u_0$ , and  $u_1$  are the same as given in (8), and

$$\begin{aligned}
 \delta_0 &= \sum_{i=0}^{t-1} (i - u_0)^2 \times \frac{P_i}{w_0}, \\
 \delta_1 &= \sum_{i=t}^{L-1} (i - u_1)^2 \times \frac{P_i}{w_1}, \\
 \delta_0 &= \sum_{i=0}^{L-1} (i - u)^2 \times \frac{P_i}{w}, \\
 w &= \sum_{i=0}^{L-1} P_i, \\
 u &= \sum_{i=0}^{L-1} i \times \frac{P_i}{w}.
 \end{aligned} \tag{10}$$

Expanding this logic to multilevel threshold,

$$f_{12}(t) = w_0 \times \delta_0^2 + w_1 \times \delta_1^2 + w_2 \times \delta_2^2 + \dots + w_N \times \delta_N^2, \tag{11}$$

where  $N$  is the number of thresholds.

Equation (11) seems very similar to the uniformity measure expression and can be regarded as the objective function to be optimized for the MABC.

**5.2. Experiment Setup.** The experimental evaluations for segmentation performance by MABC are carried out on a wide variety of image datasets. These datasets involve a set of popular tested images used in previous studies [35–37], namely, avion.ppm, house.ppm, lena.ppm, peppers.ppm, safari04.ppm, and hunter.pgm (available at <http://decsai.ugr.es/cvg/dbimagenes/>). The size of each involved image is  $512 \times 512$ . A comparison between the proposed algorithm and other algorithms is evaluated by the traditional Otsu methods. The parameters of these algorithms (namely, MABC, ABC, PSO, EGA, and CMA-ES) are set as described in Section 4.1, which achieved the best results. The population-based algorithms used for image segmentation, including the MABC, may only accelerate the velocity of computation, but they fail to improve the optimal result. Therefore, we strived to utilize the proposed algorithm to obtain multiple thresholds with larger fitness values and fast computation ability.

The numbers of thresholds  $M - 1$  for image segmentation are 2, 3, 4, 5, 7, and 9. The population size of each involved algorithm is 50. The maximum number of FEs is 2000, which has sufficient time to search on high-dimension problems. The run time of each algorithm on each image for each  $M - 1$  value is 30. Figure 8 presents four original images and their histograms.

TABLE 10: Objective value and standard deviation by the compared population-based methods on Otsu algorithm.

Image	$M - 1$	Objective values (standard deviation)				
		MABC	ABC	PSO	CMA-ES	EGA
Avion	2	$3.5214E + 04$	$3.4845E + 04$	$3.4951E + 04$	$3.5212E + 04$	$3.4845E + 04$
		$3.0608E - 01$	$7.8476E - 12$	$1.6680E - 02$	$1.4223E - 01$	$6.7840E - 02$
	3	$3.5289E + 04$	$3.4923E + 04$	$3.5021E + 04$	$3.5290E + 04$	$3.4922E + 04$
		$1.6097E - 01$	$1.9359E - 01$	$1.1615E - 01$	$2.7709E - 01$	$2.7985E - 01$
	4	$3.5312E + 04$	$3.4968E + 04$	$3.5112E + 04$	$3.5336E + 04$	$3.4966E + 04$
		$4.8025E - 01$	$4.7856E - 01$	$8.0976E - 01$	$5.1612E - 01$	$1.6168E + 00$
House	2	$2.9036E + 04$	$2.8721E + 04$	$2.8836E + 04$	$2.9053E + 04$	$2.8721E + 04$
		$2.9983E - 02$	0	$6.3446E - 02$	$0.0000E + 00$	$1.7600E - 01$
	3	$2.9169E + 04$	$2.8848E + 04$	$2.8962E + 04$	$2.9180E + 04$	$2.8847E + 04$
		$2.8244E - 02$	$3.8473E - 02$	$2.0333E - 01$	$6.2417E - 03$	$6.6244E - 01$
	4	$2.9216E + 04$	$2.8897E + 04$	$2.9072E + 04$	$2.9230E + 04$	$2.8895E + 04$
		$8.4547E - 01$	$4.9432E - 01$	$6.5687E - 01$	$2.0362E - 02$	$1.4834E + 00$
Lena	2	$1.8000E + 04$	$1.7751E + 04$	$1.7824E + 04$	$1.8006E + 04$	$1.7751E + 04$
		$9.5169E - 03$	$3.9238E - 12$	$4.6357E - 02$	$3.9239E - 12$	$1.1798E - 01$
	3	$1.8136E + 04$	$1.7922E + 04$	$1.8065E + 04$	$1.8177E + 04$	$1.7920E + 04$
		$7.6749E - 03$	$5.7300E - 02$	$5.4778E - 01$	$6.4464E - 03$	$1.1566E + 00$
	4	$1.8212E + 04$	$1.7986E + 04$	$1.7983E + 04$	$1.8241E + 04$	$1.7981E + 04$
		$2.0958E - 01$	$8.1029E - 01$	$1.9688E + 00$	$4.4224E - 01$	$1.9686E + 00$
Peppers	2	$1.7603E + 04$	$1.7378E + 04$	$1.7431E + 04$	$1.7626E + 04$	$1.7379E + 04$
		$3.5888E - 01$	$3.9238E - 12$	$7.8387E - 02$	$3.9239E - 12$	$1.9094E - 01$
	3	$1.7776E + 04$	$1.7554E + 04$	$1.7679E + 04$	$1.7801E + 04$	$1.7553E + 04$
		$8.2777E - 01$	$1.4734E - 02$	$2.0108E - 01$	$2.2511E - 03$	$9.1723E - 01$
	4	$1.7852E + 04$	$1.7618E + 04$	$1.7820E + 04$	$1.7866E + 04$	$1.7614E + 04$
		$7.9962E - 01$	$8.7997E - 02$	$1.3612E + 00$	$3.9239E - 12$	$2.9410E + 00$
Safari04	2	$2.0744E + 04$	$2.0471E + 04$	$2.0667E + 04$	$2.0744E + 04$	$2.0471E + 04$
		$4.5057E - 01$	0	$4.2877E - 02$	$3.0492E - 02$	$8.2779E - 02$
	3	$2.0814E + 04$	$2.0560E + 04$	$2.0695E + 04$	$2.0834E + 04$	$2.0560E + 04$
		$7.6268E - 01$	$3.1618E - 02$	$1.8348E - 01$	$3.9239E - 12$	$5.8815E - 01$
	4	$2.0856E + 04$	$2.0594E + 04$	$2.0791E + 04$	$2.0870E + 04$	$2.0595E + 04$
		$5.6743E - 01$	$9.6848E - 01$	$1.6967E - 01$	$5.1469E - 02$	$9.0413E - 01$
Hunter	2	$9.3778E + 03$	$9.3318E + 03$	$9.3415E + 03$	$9.4923E + 03$	$9.3315E + 03$
		$1.1083E - 01$	$0.0000E + 00$	$1.8399E - 01$	$0.0000E + 00$	$5.0323E - 01$
	3	$9.5910E + 03$	$9.4838E + 03$	$9.4975E + 03$	$9.6451E + 03$	$9.4823E + 03$
		$1.2344E + 00$	$6.8290E - 01$	$4.2345E - 01$	$1.4223E - 02$	$1.3290E + 00$
	4	$9.6343E + 03$	$9.5379E + 03$	$9.5594E + 03$	$9.7024E + 03$	$9.5381E + 03$
		$8.2664E - 01$	$3.3035E + 00$	$9.8556E - 01$	$1.2903E - 01$	$1.7687E + 00$

**5.3. Experimental Results of Multilevel Threshold.** Due to the exhaustive search characteristic of classical Otsu method, its result can be regarded as the basis for comparison with other EA-based algorithms. We utilized (11) as the fitness function to evaluate involved algorithms.

*Case 1* (multilevel threshold results on test images with  $M - 1 = 2, 3, 4$ ). Table 8 presents the fitness function values, mean computation time, and corresponding optimal thresholds (with  $M - 1 = 2, 3, 4$ ) obtained by Otsu. Due to the fact that when  $M - 1 > 4$  Otsu needs to take too long time to compute, the correlative CPU times are not listed in our experiment.

However, other results attained by the population-based algorithms were compared. It is noteworthy that the real-time applications demand a suitable balance of running time and high accuracy; the term CPU times of the population-based methods should be analyzed in detail. As shown in Table 9, there are not obvious differences about CPU times between the involved population-based methods, which means that these methods based on SI and EA algorithms are suitable to deal with the high-time-complexity image segmentation scenarios.

As can be seen from Table 10, the proposed MABC algorithm generally performs close to the Otsu method in



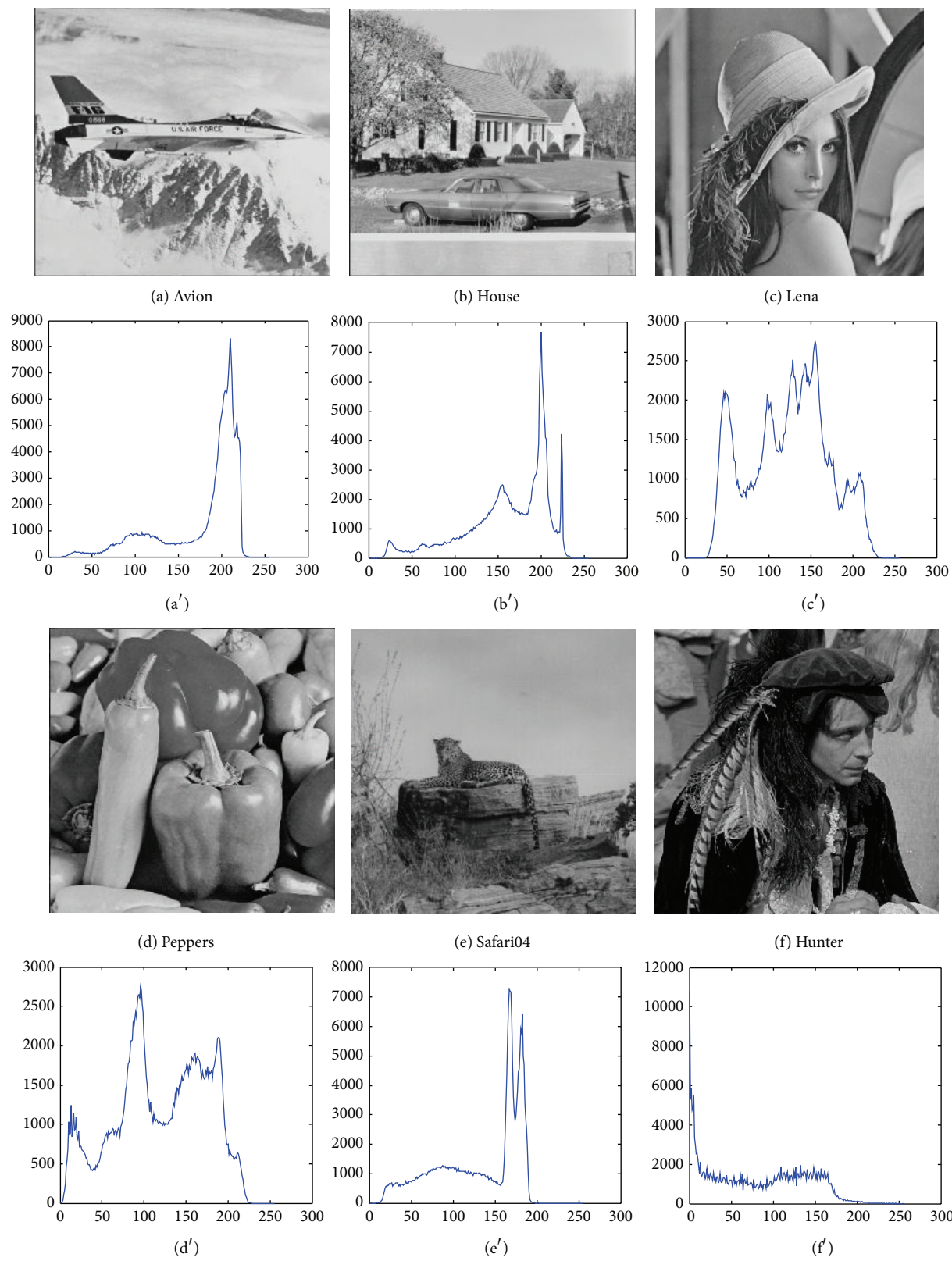
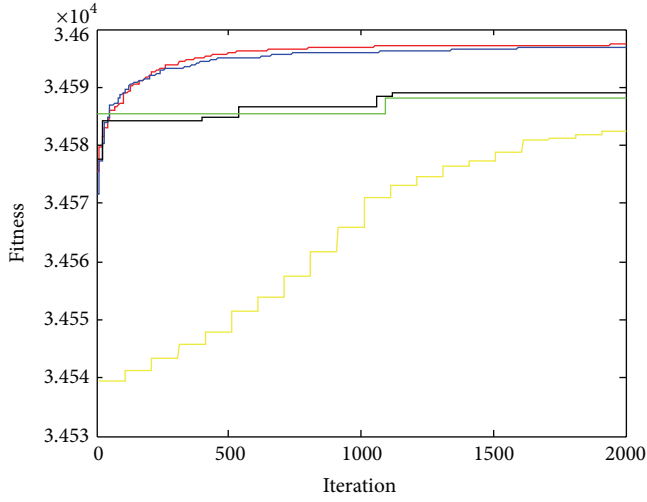
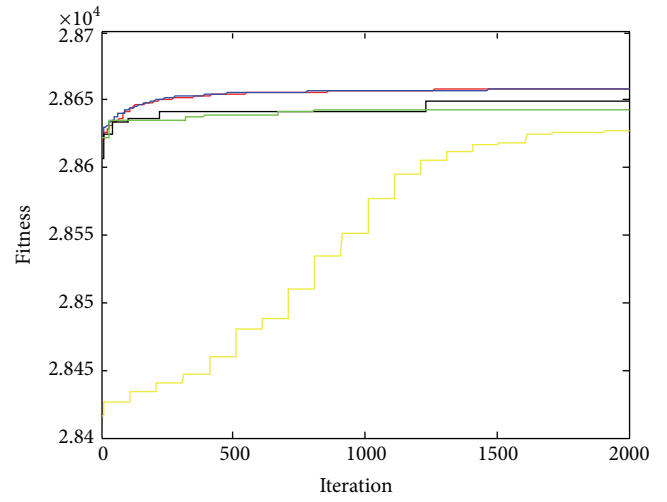


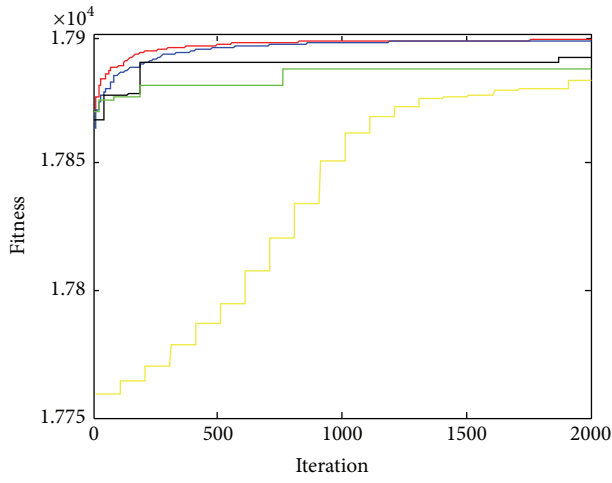
FIGURE 8: Test images and their histograms.



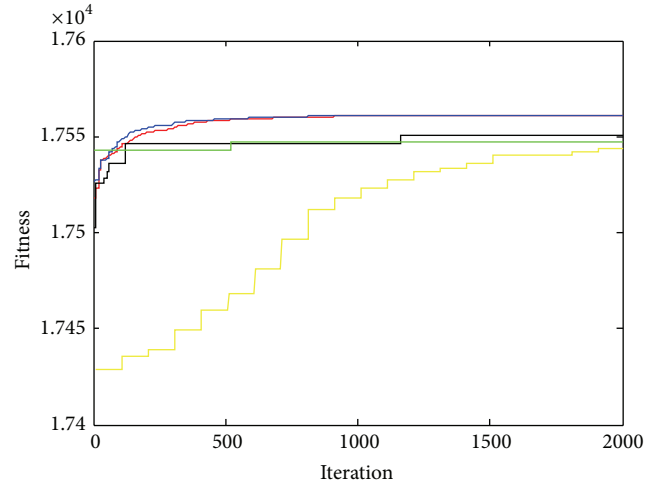
(a) Avion



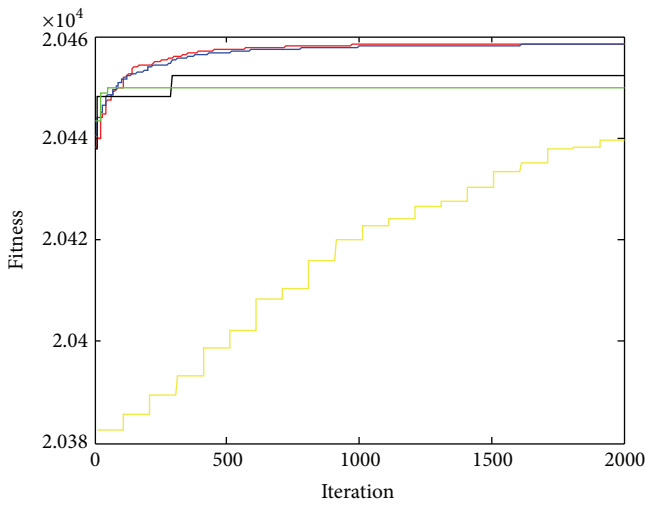
(b) House



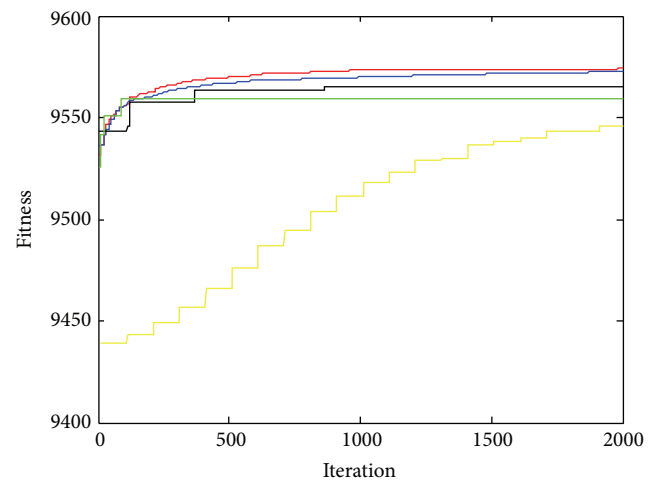
(c) Lena



(d) Peppers



(e) Safari04

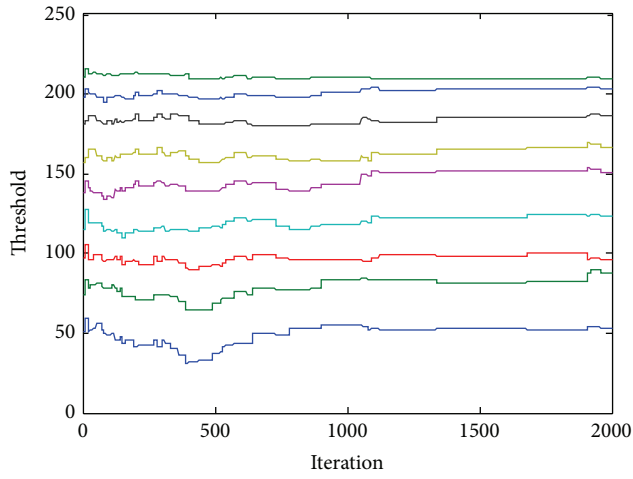


(f) Hunter

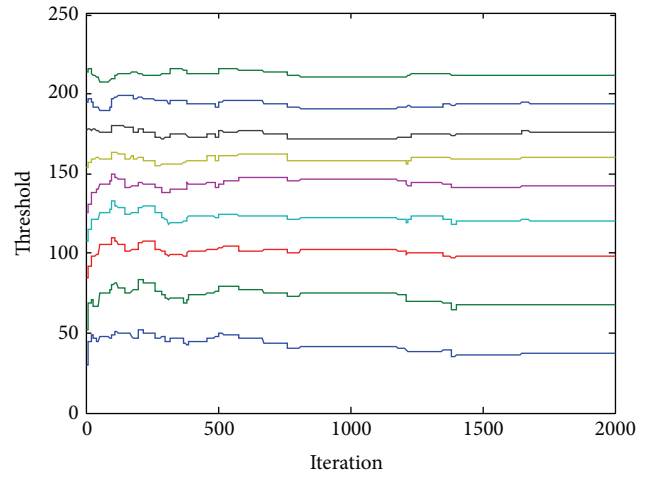
— MABC  
— ABC  
— PSO  
— EGA  
— CMA-ES

— MABC  
— ABC  
— PSO  
— EGA  
— CMA-ES

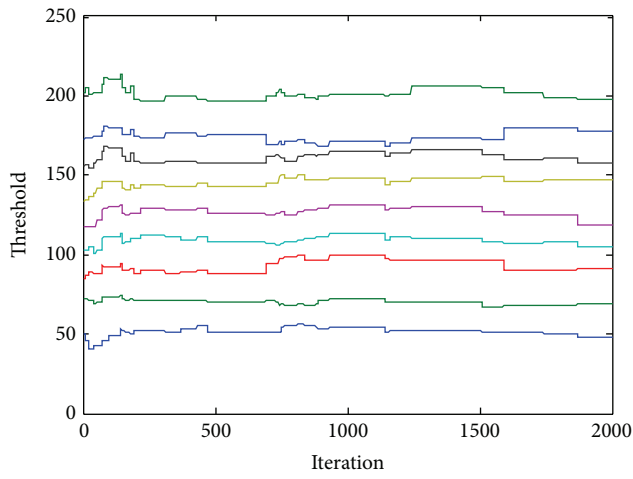
FIGURE 9: Fitness versus iteration graph by all population-based methods ( $M - 1 = 9$ ).



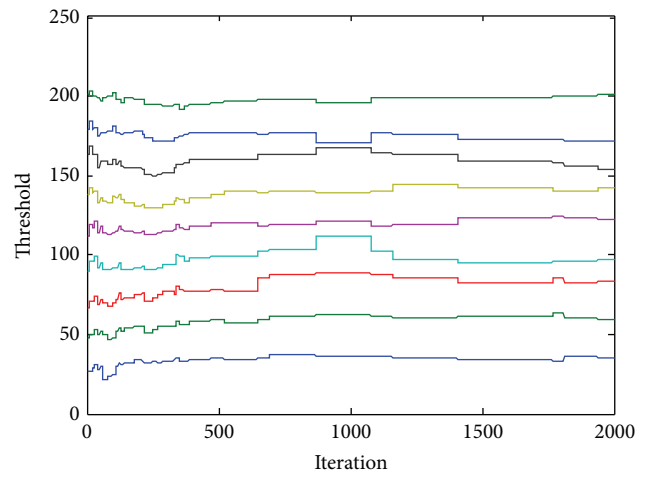
(a) Avion



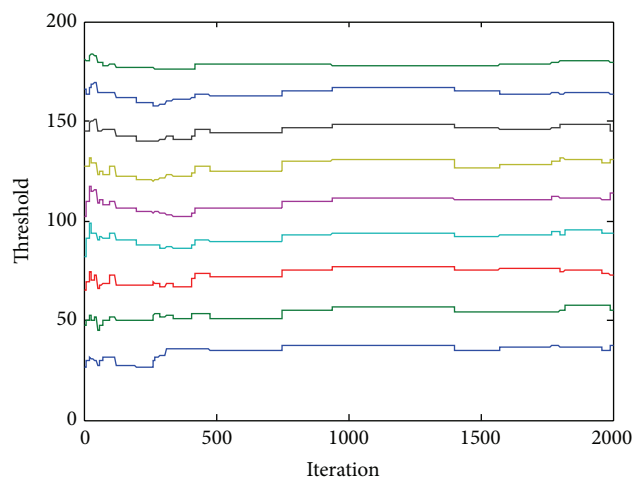
(b) House



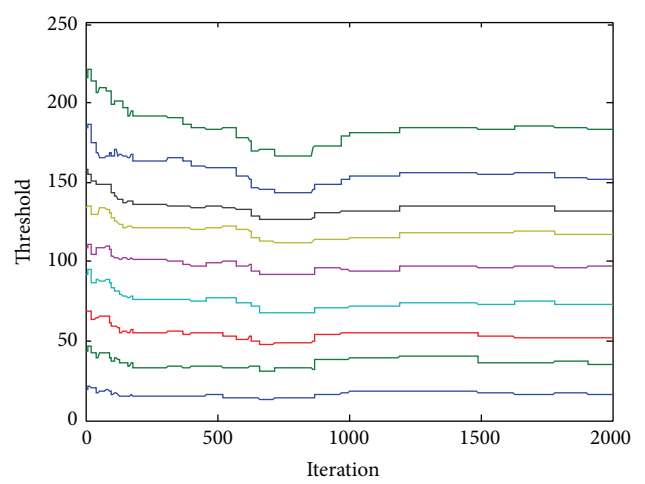
(c) Lena



(d) Peppers



(e) Safari04



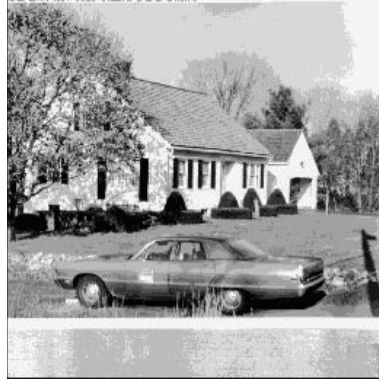
(f) Hunter

FIGURE 10: The threshold versus iteration graph by all population-based methods ( $M - 1 = 9$ ).





(a) Avion MABC



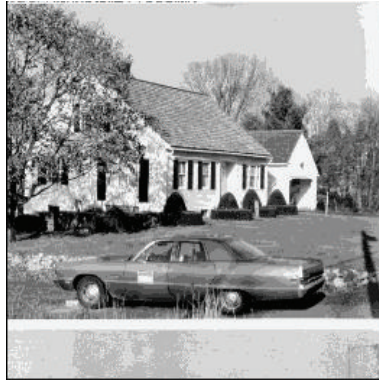
(b) House MABC



(c) Lena MABC



(a') Avion ABC



(b') House ABC



(c') Lena ABC



(a) Hunter MABC



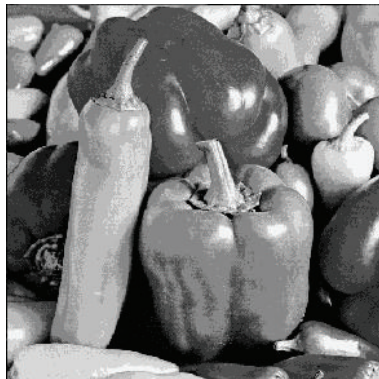
(b) Peppers MABC



(c) Safari04 MABC



(a') Hunter ABC



(b') Peppers ABC



(c') Safari04 ABC

FIGURE 11: The comparison of the thresholded images by MABC-based and ABC-based algorithms ( $M - 1 = 9$ ).

TABLE 11: Objective value and standard deviation by the compared population-based methods on Otsu algorithm.

Image	$M - 1$	Objective values (standard deviation)				
		MABC	ABC	PSO	CMA-ES	EGA
Avion	5	$3.5452E + 03$	$3.6819E + 04$	$3.5245E + 04$	$3.5367E + 04$	$3.4998E + 04$
		$5.5183E - 02$	$1.3572E + 00$	$1.0307E + 00$	$6.7539E - 02$	$2.2579E + 00$
	7	$3.5379E + 03$	$3.6841E + 04$	$3.5256E + 04$	$3.5392E + 04$	$3.5024E + 04$
		$2.5438E - 02$	$3.4426E + 00$	$1.3616E + 00$	$4.2294E - 01$	$1.3603E + 00$
	9	$3.5383E + 03$	$3.6854E + 04$	$3.5177E + 04$	$3.5404E + 04$	$3.5035E + 04$
		$1.0988E - 01$	$2.3578E + 00$	$4.5266E - 01$	$1.1011E + 00$	$1.1687E + 00$
House	5	$2.9345E + 03$	$3.0441E + 04$	$2.9175E + 04$	$2.9266E + 04$	$2.8935E + 04$
		$4.4489E - 01$	$1.6964E + 00$	$4.7929E - 01$	$4.1445E - 02$	$1.0858E + 00$
	7	$2.9357E + 03$	$3.0478E + 04$	$2.9172E + 04$	$2.9310E + 04$	$2.8972E + 04$
		$8.7026E - 02$	$4.9154E + 00$	$1.4203E + 00$	$8.1692E - 01$	$2.6283E + 00$
	9	$2.9400E + 03$	$3.0499E + 04$	$2.9092E + 04$	$2.9326E + 04$	$2.8990E + 04$
		$1.1902E - 01$	$3.4548E + 00$	$1.2017E + 00$	$8.3933E - 01$	$2.2270E + 00$
Lena	5	$1.8280E + 03$	$1.8952E + 04$	$1.8059E + 04$	$1.8269E + 04$	$1.8010E + 04$
		$9.1989E - 03$	$2.1091E + 00$	$5.7548E - 01$	$5.5638E - 01$	$1.4073E + 00$
	7	$1.8395E + 03$	$1.8983E + 04$	$1.8134E + 04$	$1.8220E + 04$	$1.8039E + 04$
		$1.0573E - 01$	$4.5723E + 00$	$7.3671E - 01$	$1.3477E - 01$	$1.9560E + 00$
	9	$1.8318E + 03$	$1.8994E + 04$	$1.8214E + 04$	$1.8315E + 04$	$1.8054E + 04$
		$1.4687E - 01$	$3.4968E + 00$	$1.4764E + 00$	$1.2792E - 01$	$2.4013E + 00$
Peppers	5	$1.8081E + 03$	$1.8584E + 04$	$1.7809E + 04$	$1.7912E + 04$	$1.7660E + 04$
		$4.1646E - 02$	$1.5773E + 00$	$1.6892E + 00$	$4.0933E - 03$	$3.0113E + 00$
	7	$1.7930E + 03$	$1.8622E + 04$	$1.7763E + 04$	$1.7951E + 04$	$1.7696E + 04$
		$1.1490E - 01$	$2.5928E + 00$	$2.1580E + 00$	$1.4511E - 01$	$2.6579E + 00$
	9	$1.7951E + 03$	$1.8636E + 04$	$1.7857E + 04$	$1.7971E + 04$	$1.7719E + 04$
		$1.9125E - 01$	$2.5459E + 00$	$1.9363E + 00$	$6.0683E - 02$	$3.0965E + 00$
Safari04	5	$2.0940E + 03$	$2.1697E + 04$	$2.0767E + 04$	$2.0901E + 04$	$2.0625E + 04$
		$9.8527E - 02$	$5.5600E + 00$	$1.3328E + 00$	$1.3033E + 00$	$3.0745E + 00$
	7	$2.1011E + 03$	$2.1719E + 04$	$2.0811E + 04$	$2.0926E + 04$	$2.0649E + 04$
		$1.1060E - 01$	$4.3328E + 00$	$1.0439E + 00$	$1.1021E - 01$	$2.0764E + 00$
	9	$2.1030E + 03$	$2.1729E + 04$	$2.0676E + 04$	$2.0935E + 04$	$2.0658E + 04$
		$9.0669E - 02$	$1.7239E + 00$	$4.8513E - 01$	$3.5683E - 01$	$1.2033E + 00$
Hunter	5	$9.6789E + 02$	$1.0075E + 04$	$9.5818E + 03$	$9.6049E + 03$	$9.5760E + 03$
		$7.8287E - 02$	$4.2570E + 00$	$1.3346E + 00$	$9.2324E - 01$	$3.0726E + 00$
	7	$9.7207E + 02$	$1.0109E + 04$	$9.6260E + 03$	$9.6491E + 03$	$9.6092E + 03$
		$1.7118E - 01$	$3.9435E + 00$	$1.5345E + 00$	$1.6460E + 00$	$3.7852E + 00$
	9	$9.7461E + 02$	$1.0125E + 04$	$9.6483E + 03$	$9.6602E + 03$	$9.6264E + 03$
		$1.3206E - 01$	$3.2571E + 00$	$3.3544E + 00$	$8.8415E - 01$	$2.7019E + 00$

terms of fitness value when  $M - 1 = 2, 3, 4$ , whereas the performance of MABC on time complexity is significantly superior to its counterpart Otsu. In other words, the MABC-based algorithm consumes less computing times than the traditional one. This is mainly due to the fact that the crossover-based social learning strategy is used in the proposed algorithm to have faster convergence speed. Furthermore, the MABC-based algorithm achieves the best achievements among the population-based methods in most cases. This can be explained by the fact that the MABC has an appropriate balance between exploration and exploitation.

Compared with other involved SI and EA algorithms, the MABC is endowed with crossover operation to facilitate better global search in complex higher dimensional search space. Moreover, the differences between the MABC and the other algorithms are more evident as the segmentation level increases.

*Case 2* (multilevel threshold results on test images with  $M - 1 = 5, 7, 9$ ). In this section, we conduct these algorithms on image segment with  $M - 1 = 5, 7, 9$ . Table 11 demonstrates the average fitness value and standard deviation achieved



TABLE 12: Objective value, standard deviation, and CPU time by the compared population-based methods and Otsu.

Image	$M - 1$	Objective values/standard deviation/CPU time					
		MABC	ABC	PSO	CMA-ES	EGA	Otsu
MRI	2	$8.1835e + 003$	$8.1831e + 003$	$8.0751e + 003$	$8.1831e + 003$	$8.0739e + 003$	$8.1839e + 003$
		$1.91738e - 012$	0	178.4632	0	13.1342	13.1342
		$2.2340E - 01$	$2.4553E - 01$	$2.3321E - 01$	$2.2943E - 01$	$2.4543E - 01$	2.6524
	3	$8.2855e + 003$	$8.2856e + 003$	$8.2543e + 003$	$8.2856e + 003$	$8.2486e + 003$	$8.2786e + 003$
		$1.91738e - 012$	0.0743	$3.203e + 001$	$1.9174e - 012$	$2.809e + 001$	$2.809e + 001$
		$3.8653E - 01$	$3.7543E - 01$	$3.7645E - 01$	$3.9853E - 01$	$4.5434E - 01$	$7.4345E + 01$
	4	$8.3433e + 003$	$8.3282e + 003$	$8.3124e + 003$	$8.3292e + 003$	$8.31908e + 003$	$8.33908e + 003$
		0	1.2580	$1.6362e + 001$	0	$1.9743e + 001$	$1.9743e + 001$
		$3.8124E - 01$	$3.8943E - 01$	$4.0126E - 01$	$3.8715E - 01$	$3.7975E - 01$	$4.5623E + 03$
	5	$8.3523e + 003$	$8.3464e + 003$	$8.3270e + 003$	$8.3503e + 003$	$8.3211e + 003$	$8.3411e + 003$
		0.70958	2.9456	$2.7728e + 001$	0.9461	$3.0005e + 001$	$3.0005e + 001$
		$3.9354E - 01$	$3.9653E - 01$	$3.9124E - 01$	$3.8210E - 01$	$4.01674E - 01$	$3.6524E + 05$
	7	$8.3763e + 003$	$8.3663e + 003$	$8.3565e + 003$	$8.3739e + 003$	$8.3499e + 003$	$8.3699e + 003$
		1.5125	1.2221	$1.0010e + 001$	$2.55323e - 002$	$1.8043e + 001$	$1.8043e + 001$
		$4.2875E - 01$	$4.2532E - 01$	$4.3215E - 01$	$4.6432E - 01$	$4.5321E - 01$	$8.6534E + 06$
	9	$8.3841e + 003$	$8.3767e + 003$	$8.3617e + 003$	$8.3841e + 003$	$8.3587e + 003$	$8.3787e + 003$
		$5.8333e - 002$	1.4733	$2.1030e + 001$	$4.9322e - 002$	$3.2366e + 001$	$3.2366e + 001$
		$5.9643E - 01$	$6.1212E - 01$	$5.94326E - 01$	$5.9942E - 01$	$6.2654E - 01$	$6.6543E + 08$

by each population-based algorithm, where the correlative results with the larger values and smaller standard deviations indicate the better achievement. Figure 9 shows the convergence curves on the test images by involved methods, and Figure 10 demonstrates the evolution curve of the optimal threshold values by MABC.

From Table 11, one can observe that there are statistically significant differences between experiments using these population-based segmentation algorithms, for both efficiency (fitness values) and stability (standard deviation). Relying on the hybrid strategies and the enhanced search ability, MABC demonstrates powerful performance and robustness on the problems with higher dimension. Compared with other state-of-the-art population-based methods except MABC, the ABC-based method has relative stronger exploration ability due to its usage of scout bees mechanism. However, as can be seen from Table 11 and Figure 9, by increasing the level of segmentation, the fitness of MABC increases more than the fitness of the other methods while ABC gives almost the same fitness for 5, 7, and 9 levels of segmentation since it is not endowed with any kind of mechanism to improve the convergence of individual when in the vicinities of the optimal solution. The results indicate that the MABC-based method can be regarded more suitable to resolve multilevel image segmentation problems. From Figure 10, it can be validated that the MABC-based method can quickly search the optimal thresholds. Segmented images based on the MABC and ABC methods with  $M - 1 = 9$  are shown in Figure 11.

Case 3 (multilevel threshold results on MRI images). To further investigate its performance on medical specific application, the utility of the proposed MABC algorithm is demonstrated by considering the T2-weighted brain magnetic resonance image (MRI) slice. The test image and its histogram are shown in Figure 12. The segmentation results of the proposed method are compared with the ABC, PSO, CMA-ES, and EGA methods. The experimental setup is referred to in Section 5.2, and the parameters of the involved algorithms are set as described in Section 4.1.

The aim of the proposed MABC algorithm is to obtain the correct threshold values and higher objective values with fast computation ability. Using the MABC algorithm and other heuristic algorithms, the objective values, the corresponding standard deviation values, and the mean computation time obtained are shown in Table 12. We note that the higher the objective value, shown in Table 12, the better the segmentation result.

From Table 12, concerning the computation time, the involved population-based methods perform closely to each other while the Otsu method needs to take too long time to compute, which validate that these population-based methods have obvious advantage to deal with the high-time-complexity image segmentation scenarios. Furthermore, the MABC method obtains the best objective fitness values among the population-based methods in most cases. Figure 13 shows the objective fitness convergence curves with  $M - 1 = 9$  obtained by the involved methods. From Figure 13,



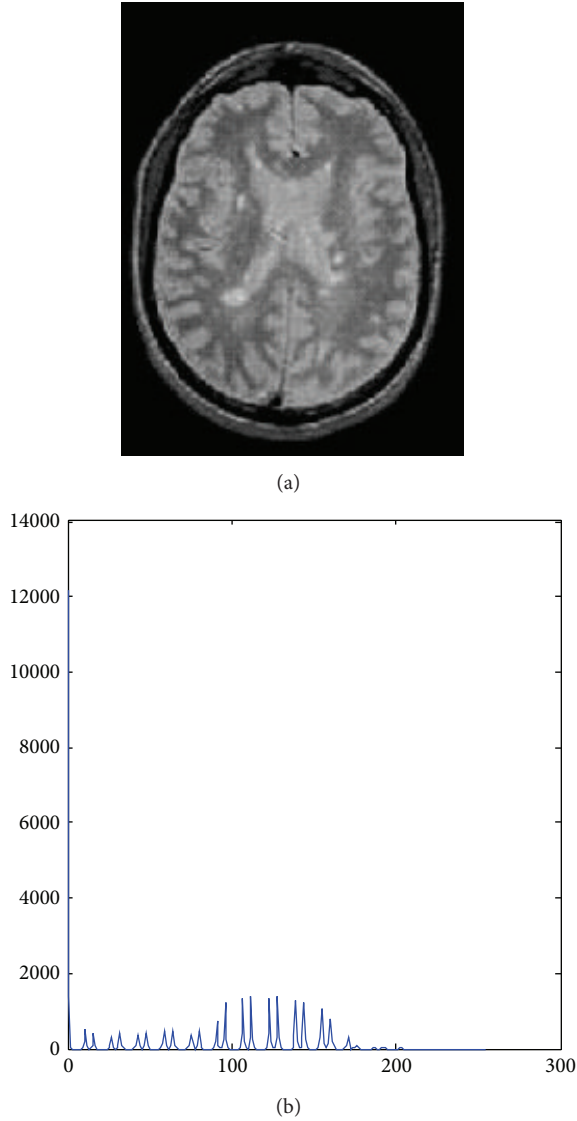


FIGURE 12: (a) The brain magnetic resonance image (MRI); (b) MRI's histogram.

we can see that the MABC performs more powerfully than its compared methods when dealing with high-dimension image segmentation. The evaluation curves of the optimal threshold values ( $M-1=9$ ) by MABC are shown in Figure 14, and the final segmented image based on the MABC and ABC methods is demonstrated in Figure 15.

## 6. Conclusions

In order to improve the ability of ABC to tackle complex image segmentation problems, this paper proposes a modified artificial bee colony optimizer by employing a pool of optimal foraging strategies, namely, MABC. The potential of the proposed MABC to enhance the search ability and maintain the diversity of population is achieved by combining several optimal foraging approaches, namely, the

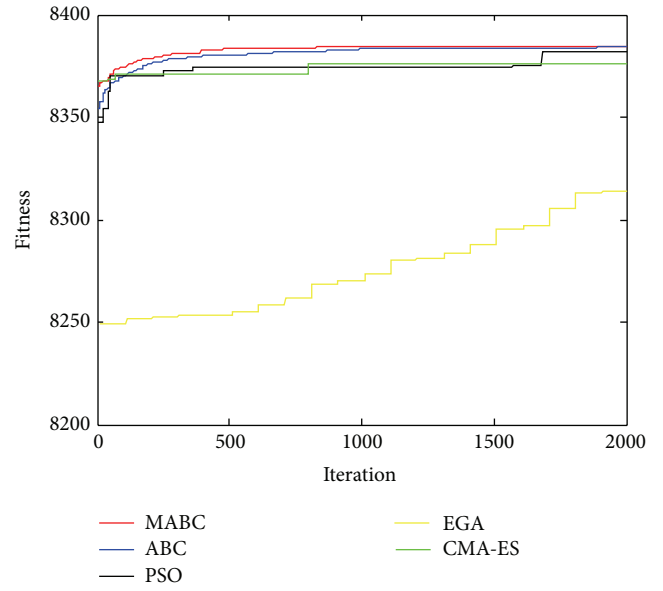


FIGURE 13: The fitness versus iteration graph by all population-based methods on MRI image ( $M-1=9$ ).

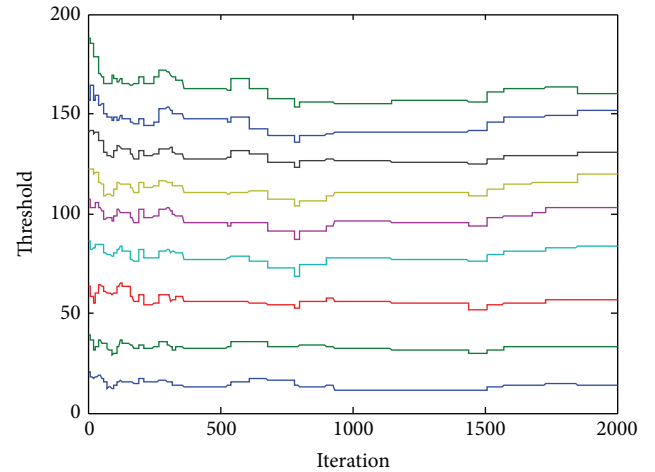


FIGURE 14: The threshold versus iteration graph by all population-based methods on MRI image ( $M-1=9$ ).

bee-to-bee communication and multipopulation cooperative mechanism.

The elaborately designed experiments are conducted on a suite of 15 benchmarks functions, to evaluate the performance of the proposed MABC in comparison with six state-of-the-art population-based algorithms. The experimental results indicate that, for most of the involved benchmarks, the MABC algorithm demonstrates more powerful performance in terms of solution accuracy, convergence speed, and robustness.

Finally, the MABC algorithm is employed to resolve the real-world image segmentation problems. The correlative results obtained by the MABC-based method on the test image dataset indicate its potential of achieving significant improvement of performance in the image segmentation

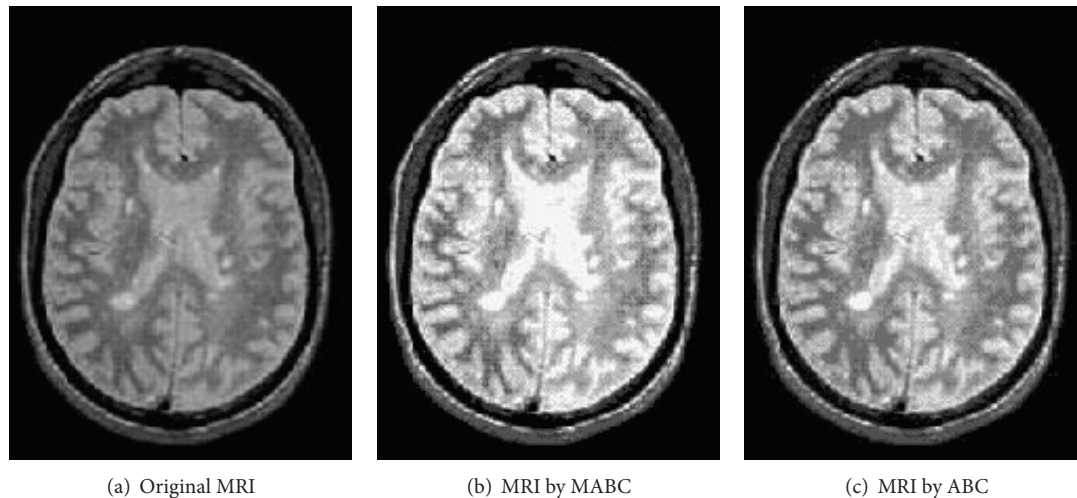


FIGURE 15: The comparison of the thresholded images by MABC-based and ABC-based algorithms ( $M - 1 = 9$ ).

application. In our future work, we will focus on designing a more efficient optimization framework based on the MABC's merits and extend its application to computer vision processing scenarios.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### References

- [1] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.
- [2] T. Pun, "Entropic thresholding, a new approach," *Computer Graphics and Image Processing*, vol. 16, no. 3, pp. 210–239, 1981.
- [3] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, Cybernet*, vol. 9, no. 1, pp. 62–66, 1979.
- [4] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.
- [5] Y. K. Lim and S. U. Lee, "On the color image segmentation algorithm based on the threshold and the fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 935–952, 1990.
- [6] D.-M. Tsai, "A fast thresholding selection procedure for multimodal and unimodal histograms," *Pattern Recognition Letters*, vol. 16, no. 6, pp. 653–666, 1995.
- [7] P. Y. Yin and L. H. Chen, "New method for multilevel threshold using the symmetry and duality of the histogram," *Journal of Electronics and Imaging*, vol. 2, pp. 337–344, 1993.
- [8] A. D. Brink, "Minimum spatial entropy threshold selection," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 142, no. 3, pp. 128–132, 1995.
- [9] H. D. Cheng, J.-R. Chen, and J. Li, "Threshold selection based on fuzzy c-partition entropy approach," *Pattern Recognition*, vol. 31, no. 7, pp. 857–870, 1998.
- [10] L.-K. Huang and M.-J. J. Wang, "Image thresholding by minimizing the measures of fuzziness," *Pattern Recognition*, vol. 28, no. 1, pp. 41–51, 1995.
- [11] A. Chander, A. Chatterjee, and P. Siarry, "A new social and momentum component adaptive PSO algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 5, pp. 4998–5004, 2011.
- [12] H. Gao, S. Kwong, J. Yang, and J. Cao, "Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation," *Information Sciences*, vol. 250, pp. 82–112, 2013.
- [13] H. Gao, W. Xu, J. Sun, and Y. Tang, "Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 4, pp. 934–946, 2010.
- [14] P. Ghamisi, M. S. Couceiro, F. M. L. Martins, and J. A. Benediksson, "Multilevel image segmentation based on fractional-order darwinian particle swarm optimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2382–2394, 2014.
- [15] E. Cuevas, D. Zaldivar, and M. Pérez-Cisneros, "A novel multi-threshold segmentation approach based on differential evolution optimization," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5265–5271, 2010.
- [16] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, Kayseri, Turkey, 2005.
- [17] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [18] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing*, vol. 4529 of *Lecture Notes in Computer Science*, pp. 789–798, 2007.
- [19] L. Ma, K. Hu, Y. Zhu, and H. Chen, "A hybrid artificial bee colony optimizer by combining with life-cycle, Powell's search and crossover," *Applied Mathematics and Computation*, vol. 252, pp. 133–154, 2015.

- [20] D. Karaboga, B. Akay, and C. Ozturk, "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks," *Modeling Decisions for Artificial Intelligence*, vol. 4617, pp. 318–329, 2007.
- [21] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 204–210, 2004.
- [22] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '02)*, pp. 1671–1676, Honolulu, Hawaii, USA, May 2002.
- [23] B. Akay and D. Karaboga, "Parameter tuning for the artificial bee colony algorithm," in *Proceeding of the 1st International Conference on Computational Collective Intelligence (ICCCI '9)*, Wroclaw, Poland, 2009.
- [24] Z. Chen, Z. He, and C. Zhang, "Image multi-level thresholds based on PSO with Von Neumann neighborhood," *Application Research of Computers*, vol. 26, no. 5, pp. 1977–1979, 2009.
- [25] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, Wiley, 2009.
- [26] H. Wolpert and W. G. Macready, "No free lunch theorems for search," Tech. Rep. SFI-TR-95-02-010, Santa Fe Institute, 1995.
- [27] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
- [28] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [29] L. Ma, K. Hu, Y. Zhu, and H. Chen, "Cooperative artificial bee colony algorithm for multi-objective RFID network planning," *Journal of Network and Computer Applications*, vol. 42, pp. 143–162, 2014.
- [30] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [31] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.
- [32] S. Sumathi, T. Hamsapriya, and P. Surekha, *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*, Springer, 2008.
- [33] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [34] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on System, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [35] W. B. Tao, H. Jin, and L. M. Liu, "Object segmentation using ant colony optimization algorithm and fuzzy entropy," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 788–796, 2007.
- [36] P.-Y. Yin, "Multilevel minimum cross entropy threshold selection based on particle swarm optimization," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 503–513, 2007.
- [37] L. Cao, P. Bao, and Z. K. Shi, "The strongest schema learning GA and its application to multilevel thresholding," *Image and Vision Computing*, vol. 26, no. 5, pp. 716–724, 2008.



