

Research Article

Hierarchical Artificial Bee Colony Optimizer with Divide-and-Conquer and Crossover for Multilevel Threshold Image Segmentation

Maowei He,^{1,2} Kunyuan Hu,¹ Yunlong Zhu,¹ Lianbo Ma,^{1,2} Hanning Chen,¹ and Yan Song¹

¹ Department of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

² University of Chinese Academy of Sciences, Beijing 100039, China

Correspondence should be addressed to Lianbo Ma; malb@sia.cn, Hanning Chen; 1464071948@qq.com and Yan Song; songyan@sia.cn

Received 11 April 2014; Revised 24 June 2014; Accepted 26 June 2014; Published 2 September 2014

Academic Editor: Zbigniew Leśniak

Copyright © 2014 Maowei He 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 presents a novel optimization algorithm, namely, hierarchical artificial bee colony optimization (HABC), for multilevel threshold image segmentation, which employs a pool of optimal foraging strategies to extend the classical artificial bee colony framework to a cooperative and hierarchical fashion. In the proposed hierarchical model, the higher-level species incorporates the enhanced information exchange mechanism based on crossover operator to enhance the global search ability between species. In the bottom level, with the divide-and-conquer approach, each subpopulation runs the original ABC method in parallel to part-dimensional optimum, which can be aggregated into a complete solution for the upper level. The experimental results for comparing HABC with several successful EA and SI algorithms on a set of benchmarks demonstrated the effectiveness of the proposed algorithm. Furthermore, we applied the HABC to the multilevel image segmentation problem. Experimental results of the new algorithm on a variety of images demonstrated the performance superiority of the proposed algorithm.

1. Introduction

Image segmentation is considered as the process of partitioning a digital image into multiple regions or objects. Among the existing segmentation methods, multilevel threshold technique is a simple but effective tool that can extract several distinct objects from the background [1–4]. This means that searching the optimal multiple thresholds to ensure the desired thresholded classes is a significantly essential issue. Several multiple thresholds selection approaches have been proposed in literatures; [3, 5, 6] proposed some methods derived from optimizing an objective function, which were originally developed for bilevel threshold and later extended to multilevel threshold [1, 2]. However, these methods based on exhaustive search suffer from a common drawback; that is, the computational complexity will rise exponentially when extended to multilevel threshold from bilevel.

Recently, to address this concerned issue, swarm intelligence (SI) algorithms as the powerful optimization tools have been introduced to the field of image segmentation owing to their predominant abilities of coping with complex nonlinear optimizations [7–10]. Akay [11] employed two successful population-based algorithms—particle swarm optimization (PSO) and artificial bee colony (ABC) algorithm to speed up the multilevel thresholds optimization. Ozturk et al. [12] proposed an image clustering approach based on ABC to find the well-distinguished clusters. Sarkar and Das [13] presented a 2D histogram based multilevel thresholding approach to improve the separation between objects, which demonstrated its high effectiveness and efficiency.

It is worth noting that due to its simple arithmetic and good robustness, the ABC-based approaches have achieved more attentions from researchers [14–18] and are widely used in various optimization problems [19–24]. However, when solving complex problems, ABC algorithm will still suffer

from the major drawbacks of being easily trapped into local minimums and loss of population diversity [20].

Comparing with other evolutionary and swarm intelligence algorithms [25–31], how to improve the diversity of swarm or overcome the local convergence of ABC is still a challenging issue in the optimization domain. Thus, this paper presents a novel hierarchical optimization scheme based on divide-and-conquer and crossover strategies, namely, HABC, to extend ABC framework from flat (one level) to hierarchical (multiple levels). Note that such hierarchical schemes have been applied in optimization algorithms [32–38]. The essential differences between our proposed scheme and others lie in the following aspects.

- (1) The divide-and-conquer strategy with random grouping is incorporated in this hierarchical framework, which can decompose the complex high-dimensional vectors into several smaller part-dimensional components that are assigned to the lower level. This can enhance the local search ability (exploitation).
- (2) The enhanced information exchange strategy, namely, crossover, is applied to interaction of different population to maintain the diversity of population. In this case, the neighbor bees with higher fitness can be chosen to crossover, which effectively enhances the global search ability and convergence speed to the global best solution (exploration).

The rest of the paper is organized as follows. Section 2 describes the canonical ABC algorithm. In Section 3 the proposed hierarchical artificial bee colony (HABC) model is given. Section 4 presents the experimental studies of the proposed HABC 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, proposed by Karaboga in 2005 [19] and further developed by Karaboga and Basturk [20, 21] for real-parameter optimization, which simulates the intelligent foraging behavior of a honeybee swarm, is one of the most recently introduced swarm-based optimization techniques.

The entire bee colony contains three groups of bees: employed bees, onlookers, and scouts. Employed bees explore the specific food sources and, meanwhile, pass the food information to onlooker bees. The number of employed bees is equal to that of food sources; in other words, each food source owns only one employed bee. Then onlooker bees choose good food sources based on the received information and then 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 indexes; ϕ 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 fitness_i is the fitness value of 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. Hierarchical Artificial Bee Colony Algorithm

3.1. Hierarchical Multipopulation Optimization Model. As shown in Figure 1, the HABC framework contains two levels to balance exploring and exploiting ability. In the bottom level, with the variables decomposing strategy, each subpopulation employs the canonical ABC method to search the part-dimensional optimum in parallel. That is, in each iteration, K subpopulations in the bottom level generate K best solutions, which are constructed into a complete solution species updated to the top level. In the top level, the multispecies community adopts the information exchange mechanism

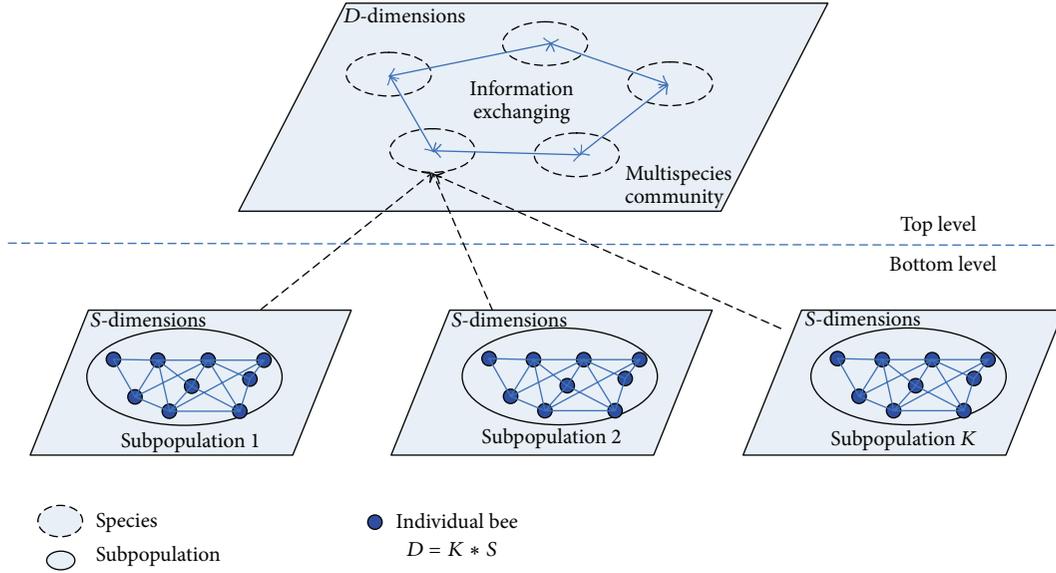


FIGURE 1: Hierarchical optimization model.

based on crossover operator, by which each species can learn from its neighborhoods in a specific topology. The vectors decomposing strategy and information exchange crossover operator can be described in detail as follows.

3.2. Variables Decomposing Approach. The purpose of this approach is to obtain finer local search in single dimensions inspired by the divide-and-conquer approach. Notice that two aspects must be analyzed: (1) how to decompose the whole solution vector and (2) how to calculate the fitness of each individual of each subpopulation. The detailed procedure is presented as follows.

Step 1. The simplest grouping method is permitting a D -dimensional vector to be split into K subcomponents, each corresponding to a subpopulation of s -dimensions, with M individuals (where $D = K * s$). The j th subpopulation is denoted as P_j , $j \in [1 \cdots K]$.

Step 2. Construct complete evolving solution $Gbest$, which is the concatenation of the best subcomponents' solutions P_j by fowling:

$$Gbest = (P_1 \cdot g, P_2 \cdot g, P_j \cdot g \cdots P_K \cdot g), \quad (4)$$

where $P_j \cdot g$ represents the personal best solution of the j th subpopulation.

Step 3. For each component P_j , $j \in [1 \cdots K]$, do the following.

- (a) *At employed bees' phase*, for each individual X_i , $i \in [1 \cdots M]$, replace the i th component of the $Gbest$ by using the i th component of individual X_i . Calculate the new solution fitness: $f(newGbest(P_1 \cdot g, P_2 \cdot g, X_i, \dots, P_K \cdot g))$. If $f(newGbest) < f(Gbest)$, then $Gbest$ is replaced by $newGbest$.

- (b) Update X_i positions by using (8).

- (c) At onlooker bees' phase, repeat (a)-(b).

Step 4. Memorize the best solution achieved so far; compare the best solution with $Gbest$ and memorize the better one.

Random Grouping of Variables. To increase the probability of two interacting variables allocated to the same subcomponent, without assuming any prior knowledge of the problem, according to the random grouping of variables proposed by [29], we adopt the same random grouping scheme by dynamically changing group size. For example, for a problem of 50 dimensions, we can define that

$$G = \{2, 5, 10, 25, 50\}, \quad (5)$$

$$K \subset G.$$

Here, if we randomly decompose the D -dimensional object vector into K subcomponents at each iteration (i.e., we construct each of the K subcomponents by randomly selecting S -dimensions from the D -dimensional object vector), the probability of placing two interacting variables into the same subcomponent becomes higher, over an increasing number of iterations [30].

3.3. The Information Exchange Mechanism Based on Crossover Operator between Multispecies. In the top level, we adopt crossover operator with a specific topology to enhance the information exchange between species, in which each species P_j can learn from its symbiotic partner in the neighborhood. The key operations of this crossover procedure are described in Figure 2.

Step 1 (select elites for the best-performing list (BPL)). Individuals from current species P_j 's neighborhood (i.e., ring topology) with higher fitness have larger probability to be

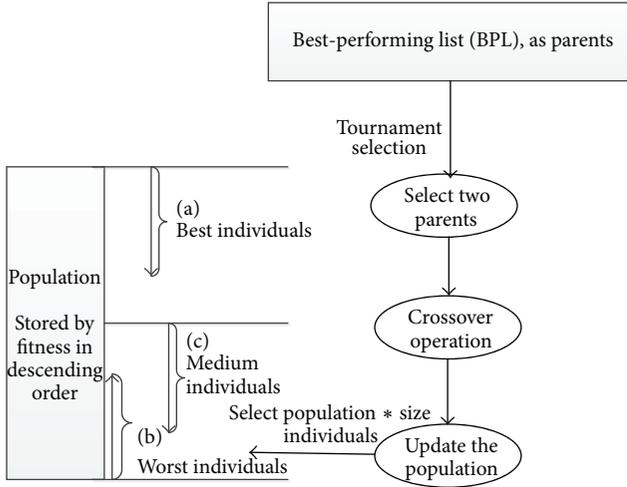


FIGURE 2: The information exchange mechanism based on crossover operation.

selected into the best-performing list (BPL) as elites, whose size is equal to current population size.

Step 2 (crossover operation)

Step 2.1. Parents are selected from the BPL's elites using the tournament selection scheme: two enhanced elites are selected randomly, and their fitness values are compared to select the elites. The one with better fitness is viewed as parent. Another parent is selected in the same way.

Step 2.2. Two offspring are created by arithmetic crossover on these selected parents by the following equation [39]:

$$s_{\text{new}} = \text{rand}(0, 1) \times \text{parent1} + \text{rand}(0, 1) \times \text{parent2}, \quad (6)$$

where S_{new} is newly produced offspring, parent 1 and parent 2 are randomly selected from BPL.

Step 3 (update with different selection schemes). If population size is S , then the replaced individuals number is $S * CR$ (CR is a selecting rate). The greedy selection mechanism is adopted to replace the selected individual. There are three replacing approaches of selecting the best individuals (i.e., $S * CR$ individuals), a medium level of individuals and the worst individuals. To maintain the population diversity, we randomly select one replacing approach at every iteration.

In summary, in order to facilitate the below presentation and test formulation, we define a unified parameters for HABC model in Table 1. According to the process description as mentioned above, the flowchart of HABC algorithm is summarized in Figure 3, while the pseudocode for HABC algorithm is presented in Pseudocode 1.

4. Experimental Study

In the experimental studies, according to the no free lunch (NFL) theorem [40], a set of eight basic benchmark functions and twelve CEC2005 benchmark functions are employed

TABLE 1: Parameters of the HABC.

HABC = $(N, M, P_{ij}, C, CR, T, O, S)$	
N	The number of species in top level
K	Subpopulation dividing D -dimensions into S -dimensions
M	Subpopulation size
S	Corresponding to a population of S -dimensions, where $S = D/K$
D	Dimensions of optimization problem
i	Top level population's (species) ID counter from 1 to N
j	Bottom level population's ID counter from 1 to K
P_{ij}	The i th population (of the j th species)
CR	Selection rate for replacing the offspring to the selected individuals.
T	The hierarchical interaction topology of HABC
O	The objective optimization goals

to fully evaluate the performance of the HABC algorithm fairly [27], as listed in the Appendix section. The number of function evaluations (FEs) is adopted as the time measure criterion substitutes the number of iterations.

4.1. Experimental Settings. Eight variants of HABC based on different crossover methods and CR values were executed with six state-of-the-art EA and SI algorithms for comparisons:

- (i) Artificial bee colony algorithm (ABC) [20];
- (ii) cooperative artificial bee colony algorithm (CABC) [41];
- (iii) canonical PSO with constriction factor (PSO) [42];
- (iv) cooperative PSO (CPSO) [28];
- (v) standard genetic algorithm (GA) [39];
- (vi) covariance matrix adaptation evolution strategy (CMA-ES) [43].

In all experiments in this section, the values of the common parameters used in each algorithm such as population size and total generation number are chosen to be the same. Population size is set as 50 and the maximum evaluation number is set as 100000. For the fifteen continuous testing functions used in this paper, the dimensions are all set as 50.

All the control parameters for the EA and SI algorithms are set to be default of their original literatures: initialization conditions of CMA-ES are the same as in [43], and the number of offspring candidate solutions generated per time step is $\lambda = 4\mu$; for ABC and CABC, the limit parameter is set to be $SN \times D$, where D is the dimension of the problem and SN is the number of employed bees. The split factor for CABC and CPSO is equal to the dimensions [28, 41]. For canonical PSO and CPSO, the learning rates c_1 and c_2 are both set as 2.05 and the constriction factor $\chi = 0.729$. For EGA, intermediate crossover rate of 0.8, Gaussian mutation rate of 0.01, and the global elite operation with a rate of 0.06 are adopted [39]. For the proposed HABC, the species number N , split factor K , and the selection rate CR should be tuned firstly in next section.

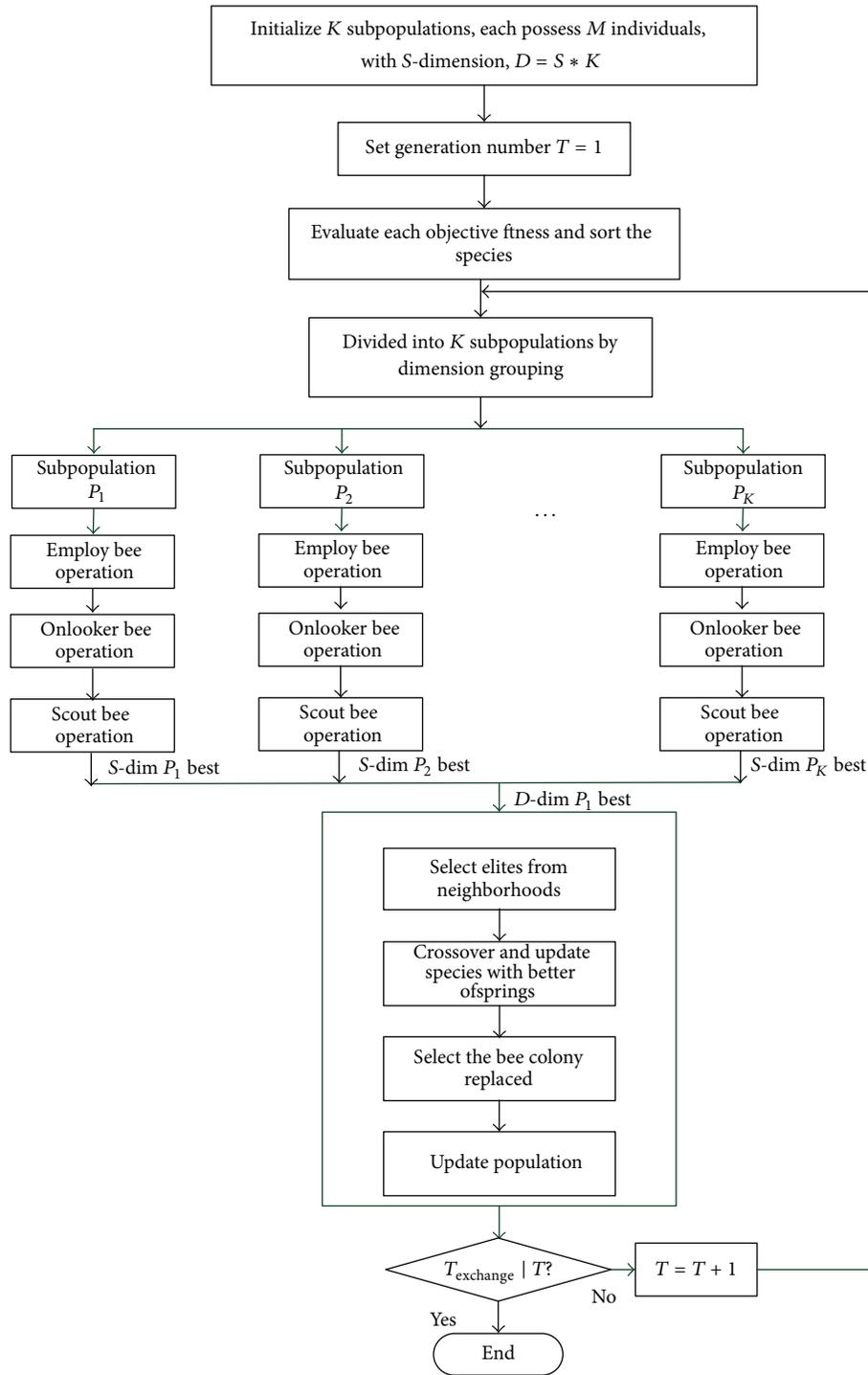


FIGURE 3: Flowchart of the HABC algorithm.

4.2. Parameter Sensitivity Analysis of HABC

4.2.1. *Effects of Species Number N.* The species number of the top level in HABC needs to be tuned. Three 50D basic benchmark functions (f_1-f_3) and five 50D benchmark functions (f_9-f_{13}) are employed to investigate the impact of this parameter. Set CR equal to 1 and all the functions run 30 sample times. As shown in Table 2, it is visible that our

proposed ABC got better optimal solutions and stability on the involved test functions with increased N. However, the performance improvement is not very remarkable using this parameter.

4.2.2. *Choices of Crossover Mode and CR Value.* The benchmark functions (f_1-f_3 and f_9-f_{13}) are adopted to evaluate

```

HABC algorithm
Set  $t := 0$ ;
INITIALIZE.
  Randomly divide the whole population into  $N$  species ( $P_i$ ) each possesses  $K$  sub-populations ( $P_{ij}$ ), each possesses  $M$  bees;
  Randomize  $P_{ij}$ 's  $D$ -dimensions food source positions  $P_{ij} \cdot X_k; i \in [1 : N], j \in [1 : K], k \in [1 : M]$ .
  Each sub-population  $P_{ij}$  with  $S$  dimensions (where  $S$  is randomly chosen from a set  $G$ , and  $D = K * S$ ).
WHILE (the termination conditions are not met)
  for each species  $i, i = [1 : N]$ 
    Initialize  $D$ -dimensions complete vector  $Gbest = (P_{i1} \cdot g, P_{i2} \cdot g, \dots, P_{i_{l-1}} \cdot g, z, P_{i_{l+1}} \cdot g \dots P_{iK} \cdot g)$ ,
    which consists of the  $S$ -dimensions best solution  $P_{i1} \hat{y}$ .
    Randomly all  $D$  dimension indices;
    WHILE (the termination conditions are not met)
      for each sub-population  $P_{ij}, j = [1 : K]$  do
        repeat
          Employed Bees' Phase:
          For each employed bee  $P_{ij} \cdot x_k$ :
            Produce a new solution by using (2)
            Evaluate the new solution
            Apply Greedy selection choosing the better solution
          end
          Calculate the probability values  $p_i$  for the solution by using (2)
          Onlooker Bees' Phase:
          for each employed bee  $P_{ij} \cdot x_k$ 
            Probabilistically choose a solution according to  $p_i$ 
            Produce a new solution by (2)
            Evaluate the new solution
            Apply Greedy selection choosing the better solution
          end
          Re-initialize solutions not improved for  $Limit$  cycles
          Memorize the best solution  $P_{ij} \cdot x$ 
          for each individual of  $P_{ij} \cdot x_k, k = [1 : M]$  do
            Place best solution in the complete solution  $newGbest$  by:
               $newGbest = (P_{i1} \cdot g, P_{i2} \cdot g, \dots, P_{ij} \cdot x_k, \dots, P_{iK} \cdot g)$ 
            Update complete solution if it improves:
              If  $f(newGbest) < f(Gbest)$ 
                Then  $P_{ij} \cdot g = P_{ij} \cdot x_k$ 
          end
        end
      end WHILE
    end WHILE
    Select elites form neighborhood of  $P_i$ 
     $BPL =$  the top  $M$  best individuals of the ring topology  $\{P_{i-1}, P_i, P_{i+1}\}$ 
    Crossover & Mutation  $P_i'$  by (4)
    Update  $P_i$  with applying Greedy selection mechanism from  $P_i'$ 
  end
  find the global best solution  $gbest$  from the whole population  $P$ 
  memorize the best solution of each  $P_{ij}$ 
  Set  $t := t + 1$ ;
end WHILE

```

PSEUDOCODE 1: Pseudocode for the HABC algorithm.

the performance of HABC variants with different crossover CRs. All the functions with 30 dimensions are implemented for 30 run times. From Table 3, we can observe that HABC variant with CR equal to 1 performed best on four functions among all five functions while CR equal to 0.05 get best result on one function. Therefore, the value of selection rate CR in each crossover operation can be set to 1 as an optimal value in all following experiments.

4.2.3. *Effects of Dynamically Changing Group Size K.* Obviously, the choice of value for split factor K (i.e., subpopulation number) had a significant impact on the performance of the proposed algorithm. In order to vary K during a run, we defined $S = \{2, 5, 10, 25, 50\}$ for 50D function optimization, and set K randomly choosing one element of S ; then, the HABC with dynamically changing K is compared with that with fixed split number on these benchmark functions for

TABLE 2: HABC's results on nine test functions with different swarm number N .

Swarm number (N)		2	6	10	14	18
f_1	Mean	0 (3.3e + 004)	0 (1.9e + 004)	0 (1.3e + 004)	0 (1.1e + 004)	0 (9.3e + 003)
	Std.	0	0	0	0	0
f_2	Mean	0 (1.9e + 004)	0 (1.2e + 004)	0 (9.0e + 003)	0 (8.4e + 003)	0 (8.0e + 003)
	Std.	0	0	0	0	0
f_3	Mean	0 (5.0e + 004)	0 (2.1e + 004)	0 (1.6e + 004)	0 (1.3e + 004)	0 (1.1e + 004)
	Std.	0	0	0	0	0
f_9	Mean	0 (4.5e + 004)	0 (3.7e + 004)	0 (2.4e + 004)	0 (2.3e + 004)	0 (2.1e + 004)
	Std.	0	0	0	0	0
f_{10}	Mean	6.43e + 002	2.32e + 002	1.38e + 001	1.31e + 001	1.29e + 001
	Std.	5.98e + 001	4.91e + 001	3.64e + 001	2.53e + 001	2.11e + 001
f_{11}	Mean	0 (5.7e + 004)	0 (4.2e + 004)	0 (3.1e + 004)	0 (2.8e + 004)	0 (2.6e + 004)
	Std.	0	0	0	0	0
f_{12}	Mean	5.84e + 003	3.13e + 003	1.08e + 003	9.31e + 002	8.73e + 002
	Std.	2.21e + 002	1.03e + 002	7.07e + 001	5.10e + 001	3.65e + 001
f_{13}	Mean	0 (3.3e + 004)	0 (1.9e + 004)	0 (1.1e + 004)	0 (9.3e + 003)	0 (8.9e + 003)
	Std.	0	0	0	0	0

TABLE 3: Results of HABC on benchmark functions (f_1 - f_3 & f_9 - f_{13}) with different CRs. In bold are the best results.

Function		0.05	0.1	0.2	0.4	0.6	1
f_1	Mean	2.34e - 109	5.65e - 215	5.32e - 290	0 (55000)	0 (45000)	0 (28000)
	Std.	4.47e - 109	0	0	0	0	0
f_2	Mean	3.97e - 004	3.12e - 006	2.65e - 005	6.87	3.12e - 006	3.54e - 007
	Std.	6.45e - 004	4.14e - 006	7.98e - 006	9.67	5.23e - 006	3.89e - 007
f_3	Mean	2.67e - 101	2.21e - 194	1.67e - 267	0 (65000)	0 (50000)	0 (35000)
	Std.	4.89e - 101	0	0	0	0	0
f_9	Mean	6.09e - 067	3.59e - 140	1.13e - 193	0 (431000)	0 (276000)	0 (181000)
	Std.	1.70e - 089	6.03e - 173	2.04e - 205	3.03e - 291	0	0
f_{10}	Mean	4.70e + 001	5.97e + 001	3.59e + 002	2.89e + 003	8.76e + 001	9.33
	Std.	3.49e + 001	2.05e + 001	7.31e + 001	6.13e + 002	4.31e + 001	6.21
f_{11}	Mean	3.11e - 120	8.14e - 091	5.43e - 103	7.53e - 073	3.19e - 099	0
	Std.	6.44e - 131	3.17e - 104	3.95e - 094	5.84e - 091	5.81e - 107	0
f_{12}	Mean	2.01e + 002	5.83e + 002	7.34e + 003	8.99e + 003	2.45e + 003	6.43e + 002
	Std.	1.57e + 001	9.39e + 001	8.73e + 001	9.33e + 002	4.31e + 002	2.51e + 001
f_{13}	Mean	2.14e - 098	5.93e - 131	2.34e - 086	4.31e - 096	5.17e - 134	0
	Std.	6.94e - 101	3.64e - 156	1.27e - 113	4.16e - 103	9.65e - 207	0

30 sample times. From the results listed in Table 4, we can observe that the performance is sensitive to the predefined K value. HABC, using a dynamically changing K value, consistently gave a better performance than the other variants except f_2 and f_{10} . Moreover, in most real-world problems, we do not have any prior knowledge about the optimal s value, so the random grouping scheme can be a suitable solution.

4.3. Comparing HABC with Other State-of-the-Art Algorithms on Benchmark Problems

4.3.1. Results on Basic Benchmark Continuous Functions. The means and standard deviations obtained all involved algorithms on the 50-dimensional classical test suite for 30

runs and were reported in Table 5, where the best results among those algorithms were shown in bold. Figures 4(a)–4(h) presented the average convergence rates of each algorithm for each basic benchmark. On the unimodal basic benchmark functions (f_1 – f_4), from Table 5 and Figure 4, HABC converged faster than all other algorithms. HABC was able to consistently find the minimum to functions f_1 , f_2 , and f_3 within 100000 FEs. Statistically, HABC has significantly superior performance on these unimodal functions. On the multimodal functions (f_5 – f_8), it is visible that HABC algorithms markedly outperforms other algorithms on most of these cases. For example, HABC quickly find the global minimum on functions f_5 and f_8 , and CABO also can consistently find the minimum of f_5 within relatively more FEs while other algorithms perform poorer. This can be

TABLE 4: Performance of HABC on 50-D (f_1 - f_3 & f_9 - f_{13}) with the different grouping number K . In bold are the best results.

Function		$K < S$	$K = 5$	$K = 10$	$K = 25$	$K = 50$
f_1	Mean	0 (95000)	0 (56000)	$3.01e - 080$	$4.43e - 043$	$7.32e - 042$
	Std.	0	0	$2.23e - 001$	$5.65e - 043$	$2.43e - 041$
f_2	Mean	$1.87e - 014$	1.58	1.87	$6.67e - 001$	1.56e - 022
	Std.	$1.65e - 014$	$3.48e - 001$	$2.38e - 001$	1.23	1.21e - 022
f_3	Mean	0	0	$4.34e - 014$	$6.87e - 013$	$9.54e - 013$
	Std.	0	0	$1.51e - 013$	$3.53e - 013$	$2.68e - 013$
f_9	Mean	0 (177000)	0 (238000)	$3.03e - 251$	$2.33e - 135$	$7.82e - 080$
	Std.	0	0	$8.03e - 301$	$3.78e - 150$	$1.10e - 103$
f_{10}	Mean	$7.15e + 001$	$2.78e + 001$	$1.58e + 001$	8.74	5.61
	Std.	$2.33e + 001$	$1.03e + 001$	5.48	$8.65e - 001$	9.91
f_{11}	Mean	0	6.93e - 210	$4.19e - 110$	$1.03e - 093$	$8.56e - 081$
	Std.	0	3.33e - 310	$3.93e - 170$	$9.51e - 104$	$5.61e - 094$
f_{12}	Mean	7.57e + 002	$2.01e + 003$	$4.78e + 003$	$7.98e + 003$	$1.13e + 004$
	Std.	5.31e + 002	$8.10e + 002$	$1.01e + 003$	$2.88e + 003$	$7.03e + 003$
f_{13}	Mean	0	$5.34e - 110$	$1.44e - 088$	$5.46e - 054$	$9.51e - 027$
	Std.	0	$3.75e - 067$	$1.39e - 045$	$3.91e - 063$	$1.93e - 010$

explained as the multipopulation cooperative coevolution strategy integrated by HABC and CABC enhance the local search ability, contributing to their better performances in the multimodal problems.

4.3.2. Results on CEC2005 Continuous Functions. To validate the effectiveness of the proposed algorithm, a suit of CEC2005 benchmarks (f_9 - f_{20}) is employed [44]. According to Section 4.2, we ensure the following optimal parameter setting of HABC: $CR = 1$, $N = 10$, and $K < S$, in comparison with CABC, CPSO, CMA-ES, ABC, PSO, and GA algorithms. From the experimental results in terms of means and standard deviations in Table 6, HABC outperformed CMA-ES on eight out of the twelve functions, CMA-ES also outperformed CABC on most functions. HABC can find the global optimum for f_9 , f_{11} , f_{13} , f_{17} , f_{18} , and f_{20} within 10000 FEs; this is due to the fact that HABC can balance the exploration and exploitation by decomposing high-dimensional problems and using crossover operations to maintain its population diversity, which is a key contributing factor. On the other hand, CMA-ES converged extremely fast. However, CMA-ES converged very fast or tended to be trapped into local minima very quickly, especially on multimodal shifted and rotated functions. According to the rank values, the performance order of the algorithms involved is HABC > CMA-ES > CABC > ABC > CPSO > PSO > GA.

In order to further investigate the efficacy and robustness of the proposed HABC, the analysis of variance (ANOVA) test was employed to determine the statistical characteristics of each proposed algorithm over the others. In this work, the graphical ANOVA analyses were done through a graphical tool of box plots, which took on many important aspects of a distribution. Through the box plot, the general features of the distribution can be noticed. The box plots shown in Figures 5(a)–5(l) demonstrate the statistical performance of each involved algorithms on the classical test suite for

30 runs individually. From this box plot representation, it is clearly visible and proved that HABC achieved good variance distribution of compromise solutions on all classical functions. Note that the CABC algorithm also exhibited its robustness on almost some classical functions.

4.3.3. Algorithm Complexity Analysis. Algorithm complexity analysis is also presented briefly as follows. If we assume that the computation cost of one individual in the HABC is $Cost_a$, the cost of the crossover operator is $Cost_c$, and the total computation cost of HABC for one generation is $N * K * M * Cost_a + N * Cost_c$. However, because the heuristic algorithms used in this paper cannot ensure comprehensive convergence, it is very difficult to give a brief analysis in terms of time for all algorithms. Through directly evaluating the algorithmic time response on different objective functions, the average computing time in 30 sample runs of all algorithms is given in Figure 6. From Figure 6, it is observed that the HABC takes the most computing time in all compared algorithms and the time increasing rate of it is the highest one. This can be explained as the multipopulation cooperative coevolution strategy integrated by HABC enhanced the local search ability at cost of increasing the computation amount. In summary, it is concluded that, compared with other algorithms, the HABC requires more computing time to achieve better results.

5. Multilevel Threshold for Image Segmentation by HABC

5.1. Entropy Criterion Based Fitness Measure. The Otsu multithreshold entropy measure [45] 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

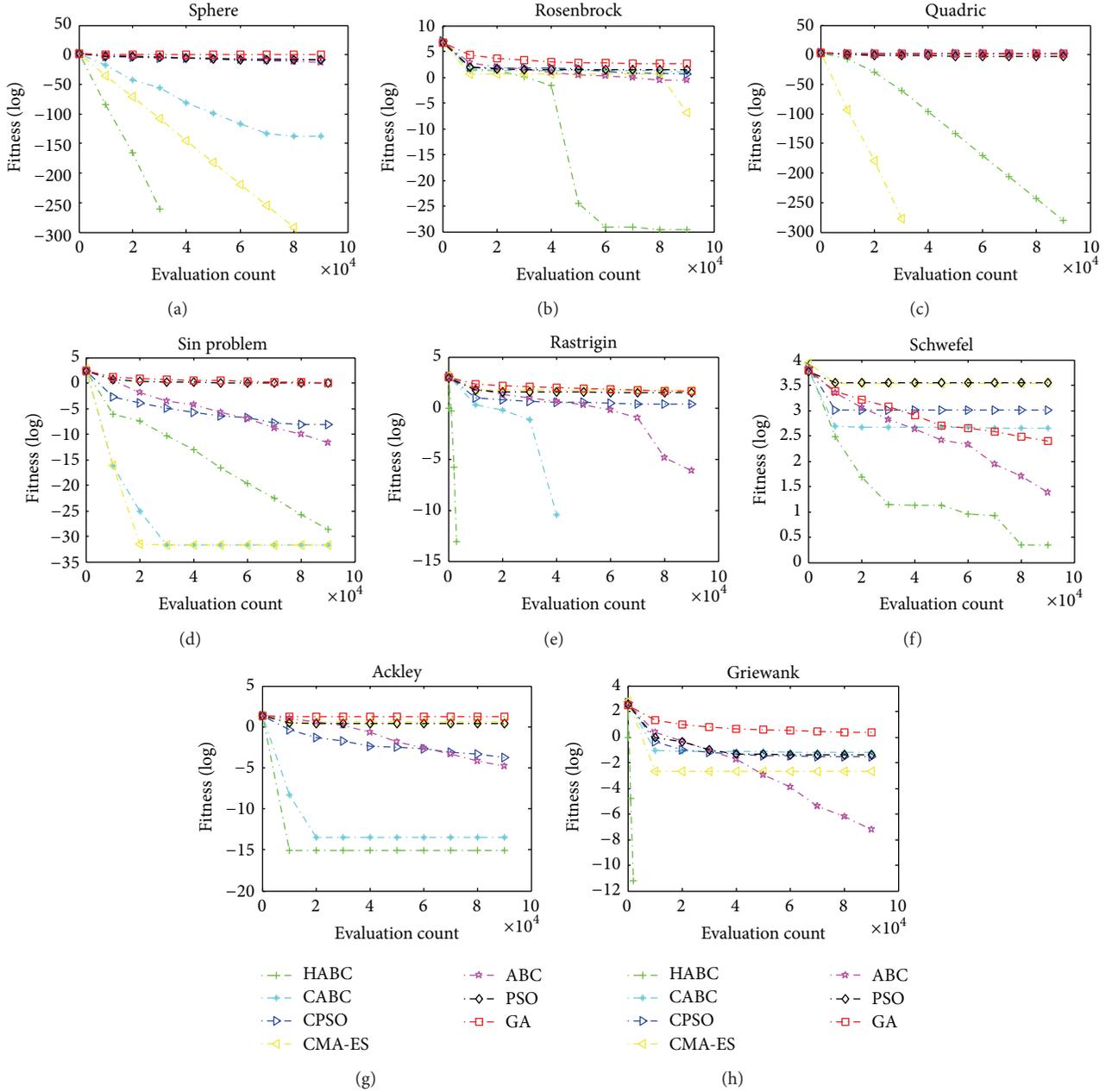


FIGURE 4: The median convergence results of 50D continuous functions. (a) Sphere. (b) Rosenbrock. (c) Quadric. (d) Sin problem. (e) Rastrigin. (f) Schwefel function. (g) Ackley. (h) Griewank.

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

$$N = \sum_{i=0}^{L-1} h(i), \quad P(i) = \frac{h(i)}{N}, \quad (7)$$

for $0 \leq i \leq L - 1$,

$$\text{Maximize } f(t) = w_0 w_1 (u_0 - u_1)^2, \quad (8)$$

where

$$w_0 = \sum_{i=0}^{t-1} P_i; \quad u_0 = \sum_{i=0}^{t-1} \frac{i \times P_i}{w_0}, \quad (9)$$

$$w_1 = \sum_{i=t}^{L-1} P_i; \quad u_1 = \sum_{i=t}^{L-1} \frac{i \times P_i}{w_1}$$

and the optimal threshold is the gray level that maximizes (8). Then, (8) can also be written as

$$f(t) = \delta^2 - w_0 \times \delta_0^2 - w_1 \times \delta_1^2, \quad (10)$$

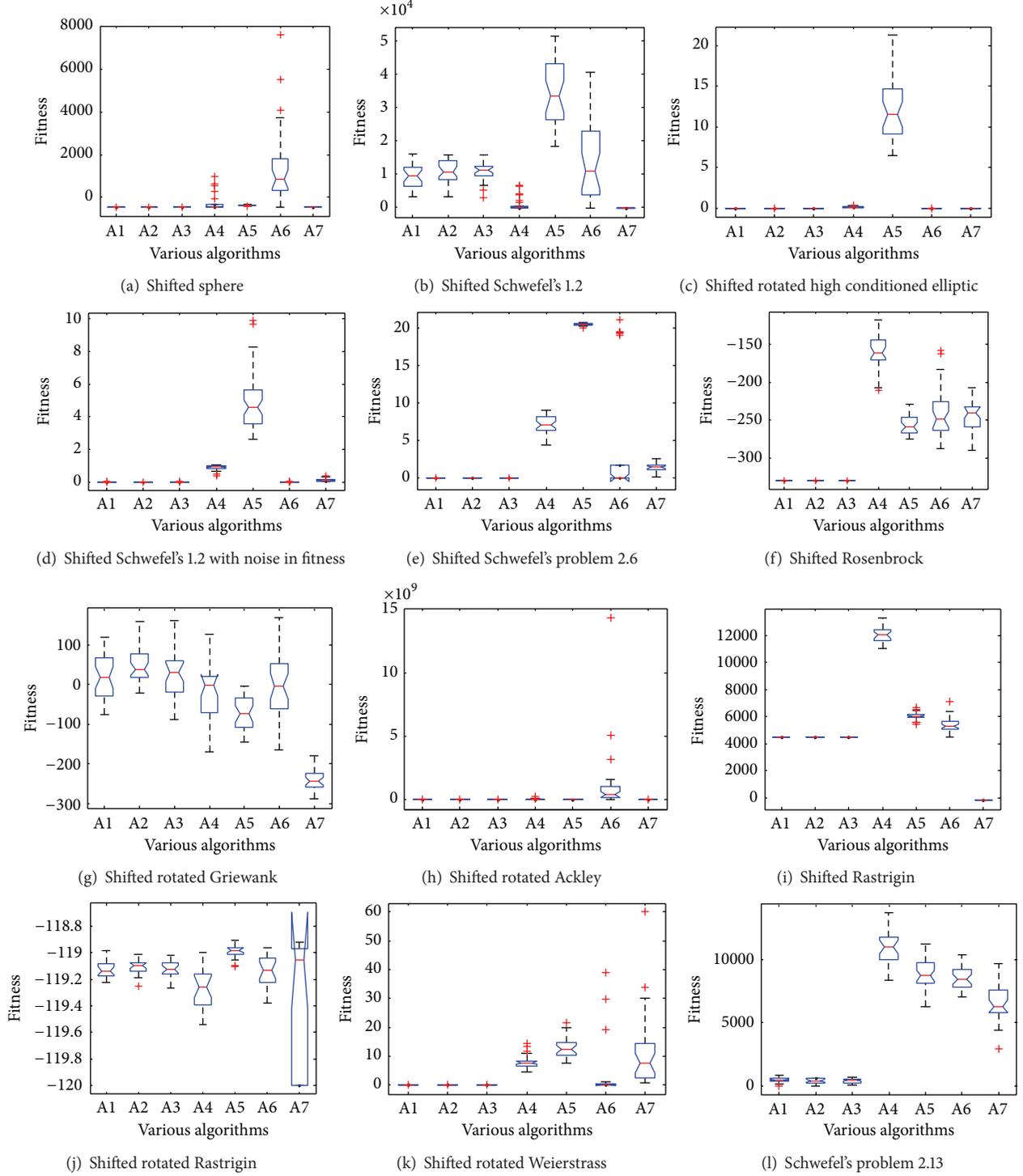


FIGURE 5: ANOVA results of all algorithms. Here, 1 to 7 are index of HABC, CABC, CPSO, CMA-ES, ABC, PSO, and GA, respectively.

where $w_0, w_1, u_0,$ and u_1 are the same as given in (9), and

$$\delta_0 = \sum_{i=0}^{t-1} \frac{(i - u_0)^2 \times P_i}{w_0},$$

$$\delta_1 = \sum_{i=t}^{L-1} \frac{(i - u_1)^2 \times P_i}{w_1},$$

$$\delta_0 = \sum_{i=0}^{L-1} \frac{(i - u)^2 \times P_i}{w},$$

$$w = \sum_{i=0}^{L-1} P_i,$$

TABLE 5: Performance of all algorithms on basic benchmark functions f_1 - f_8 .

Function		HABC	CABC	CPSO	CMA-ES	ABC	PSO	GA
f_1	Mean	0	$1.01e-137$	$3.01e-014$	0	$9.94e-016$	$1.85e-009$	$4.03e-001$
	Std.	0	$1.741e-3$	$2.45e-137$	0	$7.85e-016$	$1.01e-009$	$1.05e-001$
	Rank	1	3	5	1	4	6	7
f_2	Mean	0	4.91	3.41	$1.04e-007$	$1.83e-001$	$2.98e+001$	$3.47e+002$
	Std.	0	5.65	1.96	$4.31e-002$	$1.84e-001$	$2.48e+001$	$8.01e+002$
	Rank	1	5	4	2	3	6	7
f_3	Mean	0	$3.89e+001$	$2.66e+001$	0	$3.33e+001$	$1.99e-003$	$7.45e+001$
	Std.	0	$7.31e+001$	$4.71e+001$	0	3.97	$3.15e-003$	$3.44e+001$
	Rank	1	6	4	1	5	3	7
f_4	Mean	$3.02e-032$	$2.15e-032$	$4.97e-009$	$2.34e-032$	$8.07e-014$	$8.27e-001$	$9.10e-001$
	Std.	$4.95e-032$	$4.47e-032$	$2.08e-008$	$8.75e-034$	$5.03e-014$	$5.01e-001$	$5.11e-001$
	Rank	3	2	5	1	4	7	6
Total rank 1		6	16	18	5	16	22	27
f_5	Mean	0	0	1.35	$1.93e+001$	$6.11e-009$	$1.49e+001$	$3.92e+001$
	Std.	0	0	0.97	5.98	$5.37e-009$	$1.19e+001$	5.41
	Rank	1	1	4	6	3	5	7
f_6	Mean	2.19	$3.10e+002$	$2.32e+003$	$5.03e+003$	$1.32e+001$	$5.14e+003$	$2.79e+002$
	Std.	4.99	$1.38e+003$	$1.78e+002$	$6.08e+002$	$3.92e+001$	$6.04e+002$	$9.01e+001$
	Rank	1	4	5	6	2	7	3
f_7	Mean	$8.01e-016$	$3.15e-014$	$2.14e-004$	$1.39e+001$	$1.98e-006$	3.14	$3.92e+001$
	Std.	0	$4.75e-015$	$1.73e-004$	$6.64e-001$	$1.35e-006$	6.13	2.15
	Rank	1	2	5	7	4	6	8
f_8	Mean	0	$1.78e-008$	$4.54e-002$	$2.65e-003$	$4.94e-002$	$1.98e-002$	4.23
	Std.	0	$1.98e-008$	$3.32e-002$	$3.10e-003$	$3.95e-002$	$3.83e-002$	$3.11e-001$
	Rank	1	2	4	3	6	5	7
Total rank 2		4	9	18	22	15	23	25

$$u = \sum_{i=0}^{L-1} \frac{i \times P_i}{w}. \quad (11)$$

Expanding this logic to multilevel threshold,

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

where N is the number of thresholds.

Equation (12) is used as the objective function for the proposed HABC based procedure which is to be optimized (minimized). A close look into this equation will show that it is very similar to the expression for uniformity measure.

5.2. Probabilistic Rand Index as a Qualitative Measure. Consider a set of manually segmented (ground truth) images $\{S_1, S_2, \dots, S_K\}$ corresponding to an image $X = \{x_1, x_2, \dots, x_i, \dots, x_N\}$, where a subscript indexes one of N pixels. Let S_{test} be the segmentation that is to be compared with the manually labeled set. We denote the label of point x_i by $l_i^{S_{\text{test}}}$ in segmentation S_{test} and $l_i^{S_k}$ in the $l_i^{S_{\text{test}}}$ manually segmented image S_k . It is assumed that each label $l_i^{S_{\text{test}}}$ can take values in a discrete set of size L_k , and correspondingly $l_i^{S_{\text{test}}}$ takes one of L_{test} value.

We chose to model label relationships for each pixel pair by an unknown underlying distribution. One may visualize this as a scenario where each human segmenter provides information about the segmentation S_k of the image in the form of binary numbers $\Pi(l_i^{S_k} = l_j^{S_k})$ for each pair of pixels (x_i, x_j) . The set of all perceptually correct segmentations defines a Bernoulli distribution over this number, giving a random variable with expected value denoted as p_{ij} . Hence, the set $\{p_{ij}\}$ for all unordered pairs (i, j) defines a generative model of correct segmentations for the image X .

Consider the Probabilistic Rand Index (PRI) [46]

$$\begin{aligned} \text{PRI}(S_{\text{test}}, \{S_k\}) &= \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i < j}} [\Pi(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}}) p_{ij} + \Pi(l_i^{S_{\text{test}}} \neq l_j^{S_{\text{test}}}) (1 - p_{ij})]. \end{aligned} \quad (13)$$

Let c_{ij} denote the event of a pair of pixels i and j having the same label in the test image S_{test}

$$c_{ij} = \Pi(l_i^{S_{\text{test}}} = l_j^{S_{\text{test}}}). \quad (14)$$

TABLE 6: Performance of all algorithms on CEC 2005 functions f_9 - f_{20} .

Function		HABC	CABC	CPSO	CMA-ES	ABC	PSO	GA
f_9	Mean	0	$2.85e - 013$	$3.57e + 002$	$4.76e - 014$	$3.65e - 012$	$1.88e + 001$	$7.05e + 001$
	Std.	0	$4.62e - 014$	$3.78e + 002$	$7.14e - 014$	$2.35e - 012$	$1.98e + 001$	$4.07e + 001$
	Rank	1	3	7	2	4	5	6
f_{10}	Mean	$2.34e + 001$	$1.67e + 003$	$4.86e + 003$	0	$4.64e + 003$	$2.14e + 002$	$5.45e + 003$
	Std.	$3.86e + 001$	$3.91e + 002$	$2.24e + 004$	7.31e - 014	$4.16e + 003$	$1.01e + 002$	$2.50e + 003$
	Rank	2	4	6	1	5	3	7
f_{11}	Mean	0	$4.04e - 013$	$4.25e + 002$	$3.24e - 014$	$1.69e - 011$	$3.55e + 001$	$3.91e + 001$
	Std.	0	$6.00e - 014$	$4.84e + 002$	$2.83e - 012$	$2.96e - 012$	$6.10e + 001$	$3.90e + 001$
	Rank	1	3	7	2	4	5	6
f_{12}	Mean	$1.30e + 003$	$4.07e + 003$	$2.43e + 004$	7.65e - 022	$1.46e + 004$	$5.18e + 002$	$1.42e + 004$
	Std.	$6.77e + 001$	$7.13e + 002$	$4.42e + 005$	4.25e - 012	$4.58e + 004$	$3.64e + 002$	$8.19e + 003$
	Rank	2	4	6	1	5	3	7
f_{13}	Mean	0	$2.07e + 001$	$3.13e + 001$	0	$1.85e + 001$	$1.20e - 003$	$1.05e + 002$
	Std.	0	$5.68e + 001$	$5.44e + 001$	0	4.09	$2.31e - 003$	$2.35e + 001$
	Rank	1	6	4	1	5	3	7
f_{14}	Mean	1.37	$1.17e + 001$	$2.27e + 006$	1.72	5.72	$3.29e + 001$	$2.41e + 004$
	Std.	1.16	5.48	$1.39e + 006$	5.13	7.40	$3.27e + 001$	$1.23e + 004$
	Rank	1	4	7	2	3	5	6
f_{15}	Mean	1.66e - 003	$1.16e + 003$	$9.31e + 002$	$1.84e + 003$	$1.37e + 003$	$3.59e + 003$	$1.46e + 003$
	Std.	5.15e - 003	$2.52e - 001$	$1.90e + 001$	$2.22e - 001$	$3.78e - 008$	$3.56e + 002$	$4.31e + 001$
	Rank	1	3	5	2	3	7	6
f_{16}	Mean	$2.91e + 001$	$3.86e + 001$	$3.37e + 001$	1.51e + 001	$3.86e + 001$	$4.47e + 001$	$1.08e + 001$
	Std.	$6.75e - 002$	$2.57e - 001$	$3.39e - 002$	4.26e - 001	$1.48e - 002$	5.45	$7.51e - 003$
	Rank	3	5	2	1	5	4	7
f_{17}	Mean	0	$1.65e + 003$	$1.70e + 001$	$2.70e + 001$	$1.67e - 002$	$5.27e + 001$	$3.36e + 001$
	Std.	4.67e - 014	$1.79e + 001$	$1.66e + 001$	$4.58e + 001$	$5.02e - 002$	2.29	8.13
	Rank	1	3	4	6	2	7	5
f_{18}	Mean	2.48e + 001	$2.68e + 002$	$2.33e + 002$	$5.42e + 002$	$1.20e + 002$	$4.49e + 002$	$2.21e + 002$
	Std.	1.65e + 001	$6.44e + 001$	$9.52e + 001$	$1.42e + 001$	$1.52e + 001$	$1.96e + 001$	$2.18e + 001$
	Rank	1	3	6	2	4	5	7
f_{19}	Mean	7.99e + 001	$1.01e + 002$	$4.57e + 002$	$3.41e + 001$	$3.71e + 002$	$7.16e + 002$	$1.11e + 003$
	Std.	9.02	$2.57e + 001$	$9.03e + 001$	$1.33e + 001$	$1.83e + 001$	$1.25e + 002$	$3.34e + 002$
	Rank	2	3	5	1	4	6	7
f_{20}	Mean	3.03e + 002	$5.84e + 002$	$3.11e + 003$	$1.47e + 003$	$9.41e + 002$	$4.54e + 003$	$8.01e + 004$
	Std.	7.01e + 001	$1.05e + 002$	$6.05e + 002$	$3.93e + 002$	$2.30e + 002$	$1.01e + 003$	$1.86e + 003$
	Rank	1	2	5	4	3	6	7
Total rank		17	43	66	25	47	59	78

Then, the PRI can be written as

$$PR(S_{\text{test}}, \{S_k\}) = \frac{1}{\binom{N}{2}} \sum_{\substack{i,j \\ i < j}} [c_{ij} p_{ij} + (1 - c_{ij})(1 - p_{ij})]. \quad (15)$$

This measure takes values in $[0, 1]$, where 0 means S_{test} and $\{S_1, S_2, \dots, S_K\}$ have no similarities (i.e., when S consists of a single cluster and each segmentation in $\{S_1, S_2, \dots, S_K\}$ consists only of clusters containing single points, or vice versa) to 1 when all segmentations are identical.

5.3. Experiment Setup. The experimental evaluations for segmentation performance by HABC are carried out on

the Berkeley Segmentation Database (BSDS). The BSDS consists of 300 natural images, manually segmented by a number of different subjects. The ground-truth data for this large collection shows the diversity, yet high consistency, of human segmentation (Available at <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bds/BSDS300/html/dataset/images.html>). These datasets involve a suit of popular standard images [47–49], namely, Figures 7(a), 7(b), 7(c), 7(d), 7(e), and 7(f). The size of Figures 7(a), 7(b), 7(c), and 7(d) is $321 * 481$ and the size of Figures 7(e) and 7(f) is $481 * 321$. A comparison between the proposed algorithm and other methods is evaluated based on Otsu, which means (12) is regarded as fitness function to

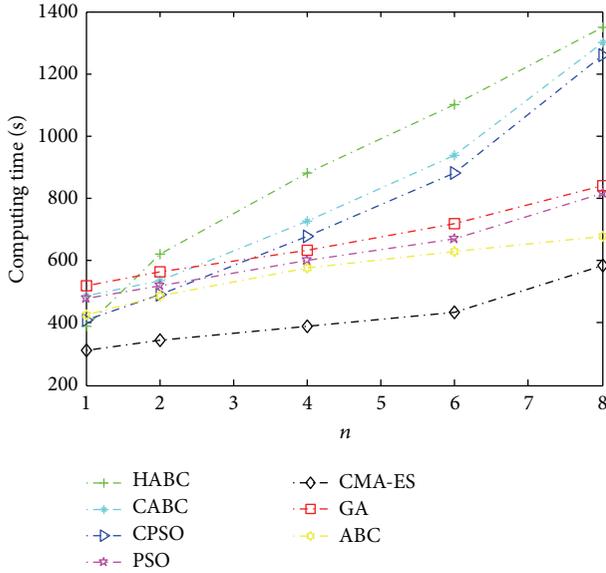


FIGURE 6: Computing time of all algorithms on different RNP problems.

evaluate all involved algorithms. The numbers of thresholds $M - 1$ investigated in the experiments were 2, 3, 4, 5, 7, and 9 while all the experiments were repeated 30 times for each image for each $M - 1$ value. The population size is 20 and the maximum number of FEs is 2000. Figure 7 donates the original images and their histograms.

The basic parameter settings of these algorithms, namely, HABC, ABC, PSO, EGA, and CMA-ES, are set as default in Section 4.1. It is noteworthy that the split factor K of HABC should be adjusted according to the dimension of the image segmentation problems. In this experiment, for 2-dimensional problem, $K \in \{1, 2\}$; for 3-dimensional problem, $K \in \{1, 3\}$; for 4-dimensional case, $K \in \{1, 2, 4\}$; for 5-dimensional case, $K \in \{1, 5\}$; for 7-dimensional case, $K \in \{1, 7\}$; for 9-dimensional case, $K \in \{1, 3, 9\}$. In order to test the effect of segmentation, the Probabilistic Rand Index (PRI) is chosen as a qualitative measure.

5.4. Experimental Results of Multilevel Threshold

Case 1 (multilevel threshold results with $M - 1 = 2, 3, 4$). Table 7 presents the fitness function values, mean computation time, and corresponding optimal thresholds (with $M - 1 = 2, 3, 4$) obtained by Otsu. It is noteworthy that the term CPU time is also an important issue in the real-time applications. From Table 8, there are not obvious differences about CPU times between the involved population-based methods, which are suitably significantly superior in terms of time complexity for high-dimensional image segmentation problems.

As can be seen from Table 9, the proposed HABC algorithm generally performs close to the Otsu method in term of fitness value when $M - 1 = 2, 3, 4$, whereas the performance of HABC on time complexity is significantly superior to its

counterpart Otsu. Furthermore, the HABC-based algorithm achieves the best performance among the population-based methods in most cases. This means that HABC can obtain an appropriate balance between exploration and exploitation. However, compared with the other population-based algorithms, the PRI results for low-dimensional segmentation obtained by HABC have no obvious enhancement. This can be explained as HABC endowed with crossover operation performed better global search in the higher-dimensional search space as well as the hierarchical cooperation strategy will be used to emphasize the fine exploitation around the promising area. Moreover, the differences between the HABC and the other algorithms are more evident as the segmentation level increases.

Case 2 (multilevel threshold results with $M - 1 = 5, 7, 9$). Regarding the high-dimensional segmentation problems with $M - 1 = 5, 7, 9$, Table 10 demonstrates the average fitness value, the standard deviation, and the PRI obtained by each population-based algorithm, where the correlative results with the larger values, the smaller standard deviations or the higher PRIs, indicate the better achievement.

From Table 10, depending on the crossover method and the fast convergence rate, HABC demonstrates the best performance in terms of efficiency and stability on the high-dimensional cases. Furthermore, as the level of segmentation increases, the fitness of HABC increases faster than that of other methods, especially ABC almost did not achieve any improvement for 5, 7, and 9 levels of segmentation. From the experimental results of Tables 8 and 9, the PRI results for high-dimensional segmentation are significantly better than that for the low-dimensional cases. Meanwhile, HABC can also achieve better statistical results regarding PRI than its counterparts in most tested datasets. This can be explained that, based on Otsu method, the segmentation with more thresholds can achieve greater consistency to human segmentation and HABC algorithm can demonstrate powerful performance in searching high-dimensional space. Figures 8, 9, 10, 11, 12, and 13 show the original images, multilevel threshold segmentations of those image, and the ground truth human segmentations of those images. From these results as shown in Figures 8–13, it is clearly visible that the HABC-based method is more suitable to deal with such multilevel segmentation problem.

6. Conclusion

In this paper, we propose a novel hierarchical artificial bee colony algorithm, called HABC, to improve the performance of solving complex problem. The main idea of HABC is to extend single artificial bee colony (ABC) algorithm to a hierarchical and cooperative mode by combining the multipopulation vector decomposing strategy and the comprehensive learning method. With the divide-and-conquer strategy, the complex vectors can be decomposed into smaller components, which are easily resolved. Through the crossover-based comprehensive learning, the information exchange between individuals can be significantly enhanced. In addition, due to lack of prior knowledge in most applications, the random



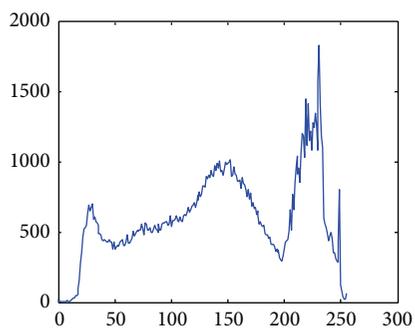
(a) 38092



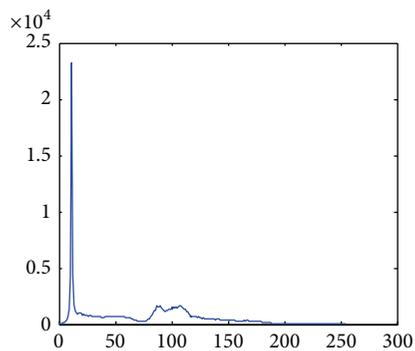
(b) 14037



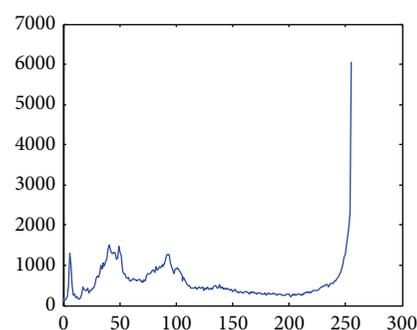
(c) 24077



(a')



(b')



(c')



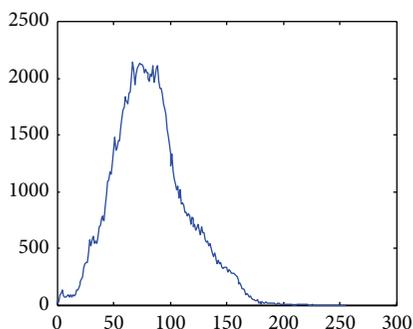
(d) 163085



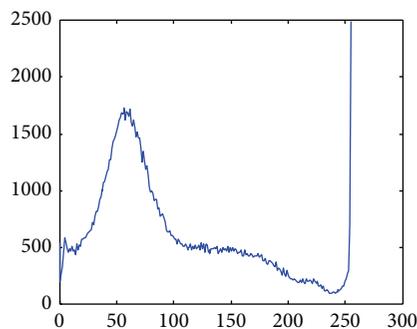
(e) 101085



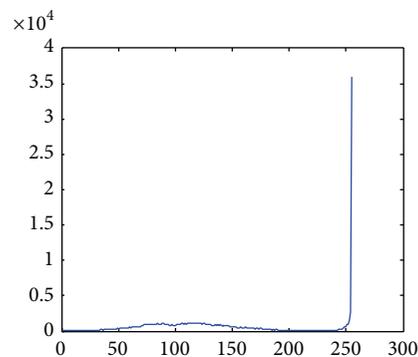
(f) 101087



(d')



(e')



(f')

FIGURE 7: Test images and their histograms.

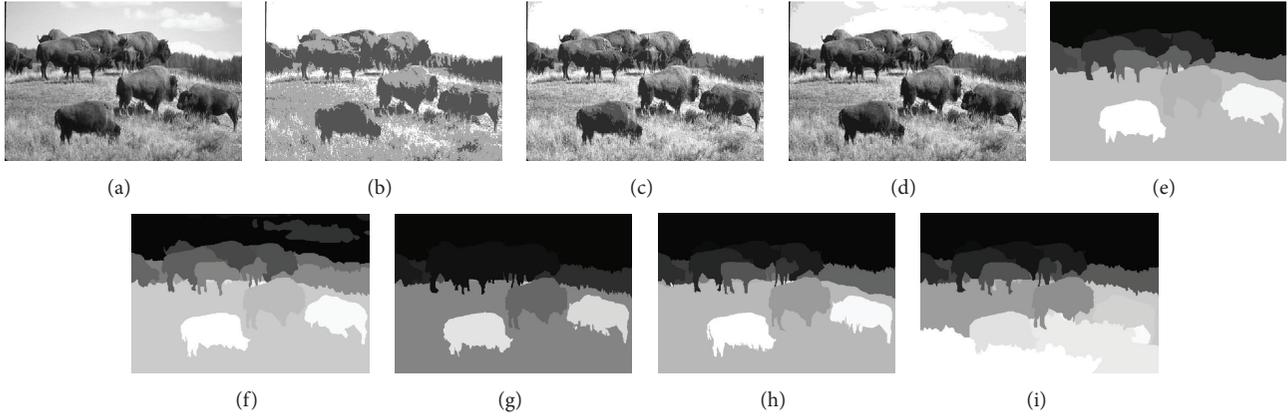


FIGURE 8: (a) Original image 38092, (b) $M - 1 = 2$ PRI = 0.74248, (c) $M - 1 = 5$ PRI = 0.80876, (d) $M - 1 = 9$ PRI = 0.79661, (e)–(i) the ground-truth hand segmentations offered by Berkeley Segmentation Dataset.

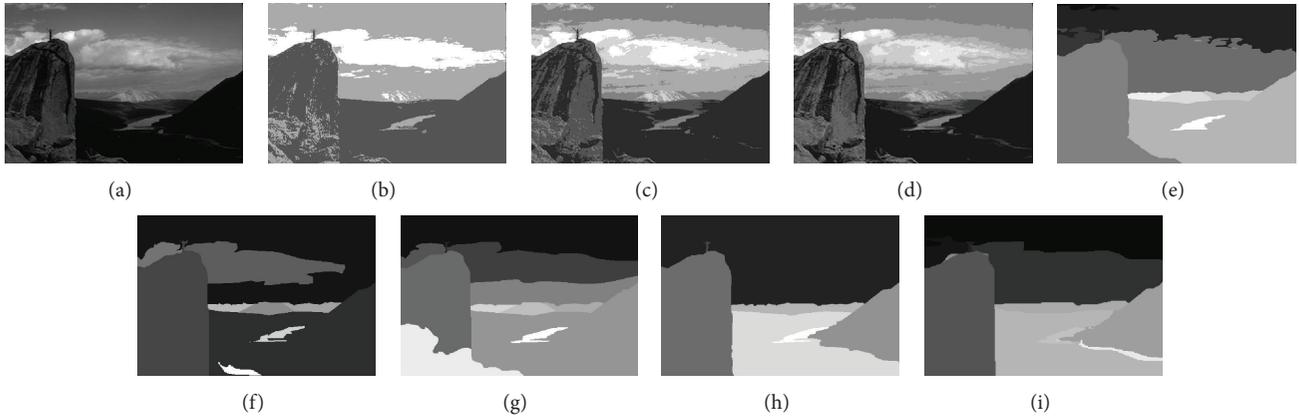


FIGURE 9: (a) Original image 14037, (b) $M - 1 = 2$ PRI = 0.73621, (c) $M - 1 = 5$ PRI = 0.79629, (d) $M - 1 = 9$ PRI = 0.80293, (e)–(i) the ground-truth hand segmentations offered by Berkeley Segmentation Dataset.

grouping technology is employed as a suitable solution. The experimental results demonstrate that the proposed HABC achieved performance superior to other classical powerful algorithms.

Finally, the HABC algorithm is applied for resolving the real-world image segmentation problems. The correlative results obtained by HABC-based method on each image indicate a significant improvement compared to several other popular population-based methods. In our future work, we aim at finding a simpler and more efficient optimization frame by using HABC's merits and then apply it to more complex image processing (up to 30 dimensions) and computer vision problems.

Appendix

A. List of Test Functions

A.1. Basic Benchmark Function

(1) *Sphere Function* is as follows:

$$f_1(x) = \sum_{i=1}^n x_i^2. \quad (\text{A.1})$$

(2) *Rosenbrock Function* is as follows:

$$f_2(x) = \sum_{i=1}^n 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2. \quad (\text{A.2})$$

(3) *Quadric Function* is as follows:

$$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2. \quad (\text{A.3})$$

(4) *Sin Function* is as follows:

$$f_4(x) = \frac{\pi}{n} \left\{ 10 \sin^2 \pi x_1 + \sum_{i=1}^{n-1} (x_i - 1)^2 (1 + 10 \sin^2 \pi x_{i+1}) + (x_n - 1)^2 \right\}. \quad (\text{A.4})$$

(5) *Rastrigin Function* is as follows:

$$f_5(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i)) + 10. \quad (\text{A.5})$$

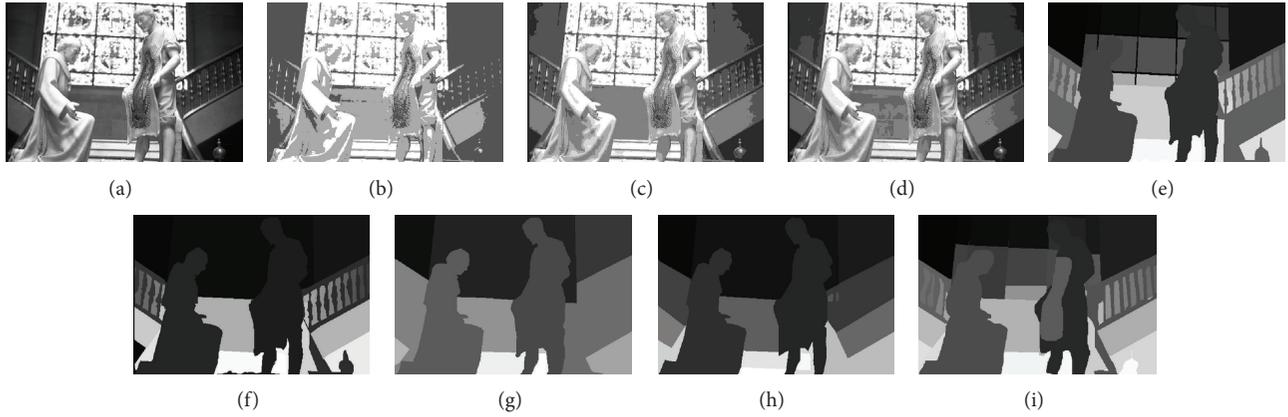


FIGURE 10: (a) Original image 24077, (b) $M - 1 = 2$ PRI = 0.68224, (c) $M - 1 = 5$ PRI = 0.79134, (d) $M - 1 = 9$ PRI = 0.84109, (e)–(i) the ground-truth hand segmentations offered by Berkeley Segmentation Dataset.

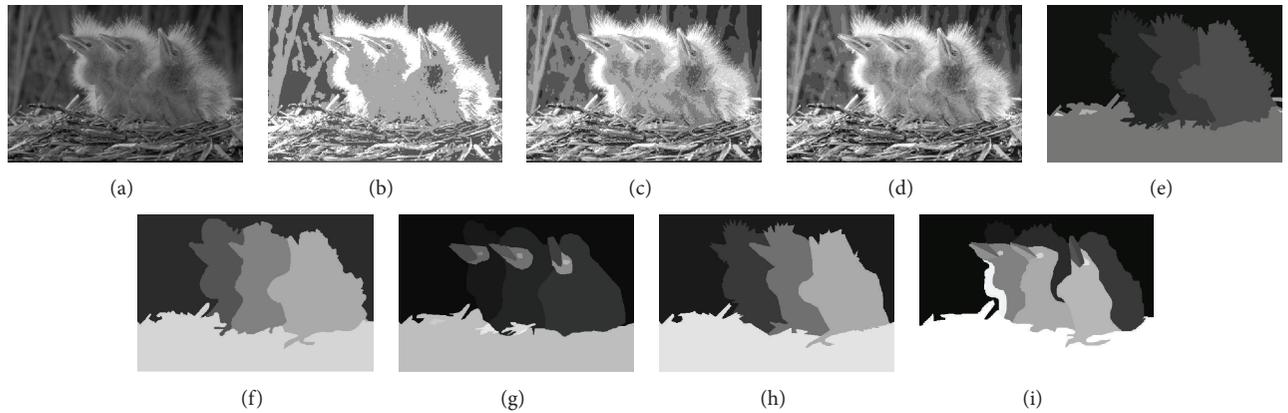


FIGURE 11: (a) Original image 163085, (b) $M - 1 = 2$ PRI = 0.61394, (c) $M - 1 = 5$ PRI = 0.67958, (d) $M - 1 = 9$ PRI = 0.70398, (e)–(i) the ground-truth hand segmentations offered by Berkeley Segmentation Dataset.

TABLE 7: Objective values and thresholds by the Otsu method and their PRI.

Image	Objective values	$M - 1 = 2$		$M - 1 = 3$			$M - 1 = 4$		
		Optimal thresholds	PRI	Objective values	Optimal thresholds	PRI	Objective values	Optimal thresholds	PRI
38092	2.5064E4	101,180	0.74248	2.5258E4	77,132,190	0.78598	25345.4	64,111,152,197	0.7994
14037	7.5361E3	61,128	0.73614	7.6593E3	33,77,131	0.8019	7.7151E4	33,73,109,147	0.79968
24077	2.1133E4	81,175	0.6821	2.1411E4	66,126,199	0.73976	2.1499E4	63,112,163,217	0.7637
163085	7.8365E3	69,110	0.61394	7.9063E3	57,85,119	0.65081	7.9426E3	52,77,102,132	0.6706
101085	1.2559E4	87,174	0.63896	1.2579E4	58,113,184	0.70822	1.2904E4	47,90,140,200	0.73810
101087	2.855E4	107,196	0.76849	2.8714E4	89,135,206	0.81706	2.8791E4	67,105,144,209	0.84213
Mean CPU time		1.4427			61.421			2644.543	

TABLE 8: The mean CPU time of the compared population-based methods on Otsu algorithm.

Dim.	Alg.				
	HABC	ABC	PSO	CMA-ES	EGA
2D	0.8575	0.7954	0.2126	0.2438	2.4755
3D	0.8956	0.8422	0.2398	0.2876	3.1928
4D	0.8982	0.8607	0.2395	0.3106	4.3953
5D	0.6322	0.8861	0.2677	0.3061	1.1138
7D	1.0136	0.9830	0.2944	0.4403	1.6599
9D	1.0461	0.9201	0.4356	0.4445	2.3235

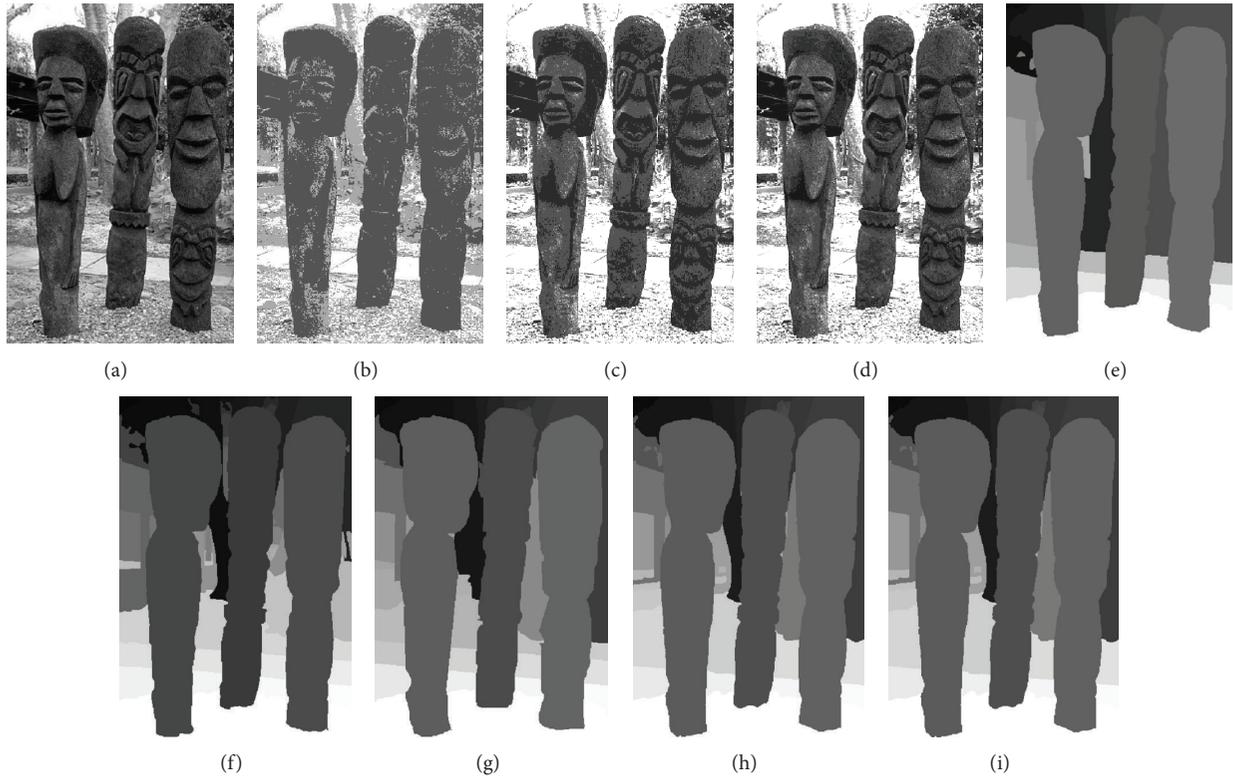


FIGURE 12: (a) Original image 101085, (b) $M - 1 = 2$ PRI = 0.63896, (c) $M - 1 = 5$ PRI = 0.76413, (d) $M - 1 = 9$ PRI = 0.80922, (e)–(i) the ground-truth hand segmentations offered by Berkeley Segmentation Dataset.

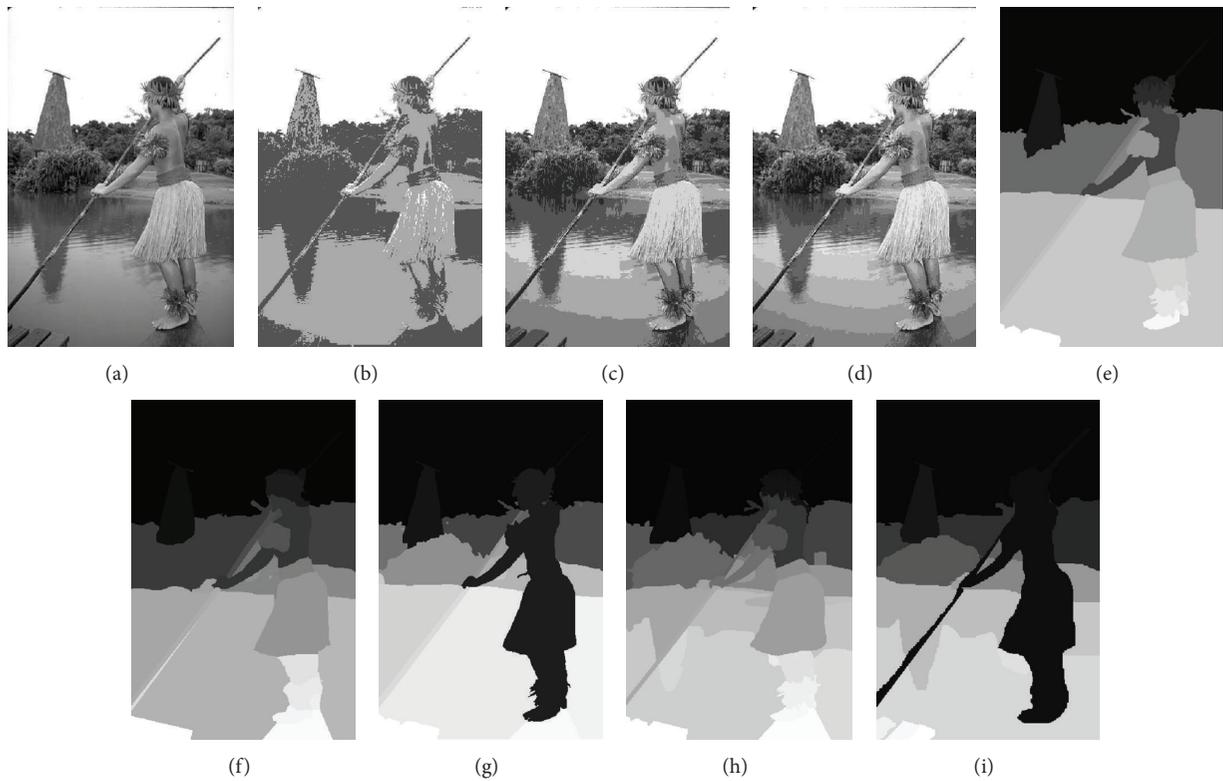


FIGURE 13: (a) Original image 101087, (b) $M - 1 = 2$ PRI = 0.76849, (c) $M - 1 = 5$ PRI = 0.85694, (d) $M - 1 = 9$ PRI = 0.88633, (e)–(i) the ground-truth hand segmentations offered by Berkeley Segmentation Dataset.

TABLE 9: Objective value, standard deviation, and PRI by the compared population-based methods on Otsu algorithm.

Image M – 1		Objective values (standard deviation)/PRI				
		HABC	ABC	PSO	CMA-ES	EGA
38092	2	2.5064E4 (3.83477E – 12)/ 0.74248	2.5064E4 (3.8348E – 12)/ 0.74248	2.5064E4 (3.8348E – 12)/ 0.74248	2.5047E4 (18.3953)/ 0.73305	2.5063E4 (0.2663)/ 0.74268
	3	2.5259E4 (0.0257346)/ 0.78598	2.5259E4 (0.0259)/ 0.78611	2.5259E4 (0.0462)/ 0.78546	2.5235E4 (39.6597)/ 0.77094	2.5256E4 (1.4455)/ 0.78657
	4	2.5345E4 (0)/ 0.7994	2.5345E4 (0.0192)/ 0.79932	2.5344E4 (0.8140)/ 0.80059	2.5276E4 (65.1685)/ 0.77711	2.5340E4 (2.0187)/ 0.79761
14037	2	7.5360E3 (0.00682035)/ 0.73621	7.5361E3 (0.0068)/ 0.73621	7.5361E3 (0)/ 0.73614	7.4730E3 (99.7498)/ 0.72324	7.5359E3 (0.2176)/ 0.73693
	3	7.6593E3 (9.58692E – 13)/ 0.8019	7.6593E3 (0)/ 0.8019	7.6592E3 (0.2069)/ 0.80121	7.6006E3 (47.7619)/ 0.76189	7.6567E3 (1.4649)/ 0.79826
	4	7.7151E3 (9.58692E – 13)/ 0.79968	7.7151E3 (1.9174E – 12)/ 0.79968	7.7120E3 (2.7500)/ 0.79803	7.6729E3 (57.4010)/ 0.78318	7.7110E3 (2.3737)/ 0.79919
24077	2	2.1133E4 (3.83477E – 12)/ 0.68224	2.1133E4 (3.8348E – 12)/ 0.68224	2.1133E4 (3.8348E – 12)/ 0.68224	2.1125E4 (7.4668)/ 0.6775	2.1133E4 (0.2056)/ 0.68234
	3	2.1411E4 (3.83477E – 12)/ 0.73976	2.1411E4 (3.8348E – 12)/ 0.73976	2.1411E4 (0.0720)/ 0.73931	2.1404E4 (9.8471)/ 0.73678	2.1409E4 (1.1128)/ 0.73934
	4	2.1499E4 (3.83477E – 12)/ 0.7637	2.1499E4 (0.0337)/ 0.76374	2.1497E4 (1.9265)/ 0.76223	2.1482E4 (11.1739)/ 0.7604	2.1495E4 (2.4823)/ 0.76263
163085	2	7.8365E3 (0)/ 0.61394	7.8365E3 (0)/ 0.61394	7.8365E3 (0)/ 0.61394	7.8156E3 (41.9717)/ 0.6188	7.8364E3 (0.1368)/ 0.6133
	3	7.9063E3 (0.0639873)/ 0.65094	7.9063E3 (0)/ 0.65081	7.9060E3 (0.7760)/ 0.64954	7.8457E3 (36.3918)/ 0.59472	7.9047E3 (1.4445)/ 0.64885
	4	7.9426E3 (0)/ 0.6706	7.9426E3 (0.0415)/ 0.67069	7.9398E3 (3.0506)/ 0.66165	7.9169E3 (21.5228)/ 0.6494	7.9402E3 (1.5929)/ 0.66836
101085	2	1.2559E4 (1.91738E – 12)/ 0.63896	1.2559E4 (1.9174E – 12)/ 0.63896	1.2559E4 (1.9174E – 12)/ 0.63896	1.2546E4 (31.8525)/ 0.63008	1.2559E4 (0.1912)/ 0.63819
	3	1.2788E4 (0.0338175)/ 0.70818	1.2788E4 (0)/ 0.70813	1.2787E4 (0.0692)/ 0.70822	1.2750E4 (54.4844)/ 0.68414	1.2785E4 (2.4630)/ 0.70652
	4	1.2904E4 (0.0274838)/ 0.73785	1.2904E4 (0.0466)/ 0.73806	1.2903E4 (1.6045)/ 0.73021	1.2885E4 (10.8032)/ 0.73136	1.2898E4 (3.2212)/ 0.73374
101087	2	2.8549E4 (0)/ 0.76849	2.8550E4 (3.8348E – 12)/ 0.76849	2.8550E4 (3.8348E – 12)/ 0.76849	2.8473E4 (186.3278)/ 0.74149	2.8549E4 (0.2269)/ 0.76852
	3	2.8715E4 (0.0071386)/ 0.81703	2.8714E4 (0.0077)/ 0.81703	2.8714E4 (0.0080)/ 0.81001	2.8684E4 (39.1601)/ 0.79675	2.8713E4 (0.9337)/ 0.81613
	4	2.8792E4 (0.0240985)/ 0.84197	2.8791E4 (7.6695E – 12)/ 0.84213	2.8791E4 (1.0312)/ 0.84132	2.8765E4 (30.3629)/ 0.83127	2.8788E4 (1.4238)/ 0.84142

(6) Schwefel Function is as follows:

$$f_6(x) = D * 418.9829 + \sum_{i=1}^D -x_i \sin\left(\sqrt{|x_i|}\right). \quad (A.6)$$

(8) Griewank Function is as follows:

$$f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \quad (A.8)$$

(7) Ackley's Function is as follows:

$$f_7(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e. \quad (A.7)$$

A.2. CEC2005 Function

(9) Shifted Sphere Function is as follows:

$$f_9(x) = \sum_{i=1}^D z_i^2 + f_{bias1}, \quad z = x - o. \quad (A.9)$$

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

Image	M - 1	Objective values (standard deviation)/PRI				
		HABC	ABC	PSO	CMA-ES	EGA
38092	5	2.5387E4 (0.130652)/ 0.80876	2.5387E4 (0.0367)/ 0.8083	2.5383E4 (2.8507)/ 0.80108	2.5347E4 (25.8857)/ 0.7798	2.5353 (1.3310)/ 0.8031
	7	2.5437E4 (0.169583)/ 0.79081	2.5436E4 (1.5625)/ 0.79184	2.5425E4 (5.1891)/ 0.78483	2.5389E4 (40.7006)/ 0.79591	2.5403E4 (1.3991)/ 0.79123
	9	2.5458E4 (0.565715)/ 0.79661	2.5458E4 (0.4417)/ 0.79642	2.5446E4 (2.5883)/ 0.79571	2.5430E4 (11.0258)/ 0.79659	2.5413 (1.0165)/ 0.79513
14037	5	7.7435E3 (0.0188984)/ 0.79629	7.7435E3 (0)/ 0.79639	7.7347E3 (3.5134)/ 0.79788	7.6961E3 (35.7684)/ 0.78357	7.7341E3 (4.3209)/ 0.79849
	7	7.7718E3 (0.261583)/ 0.80195	7.7710E3 (2.6775)/ 0.80073	7.7614E3 (4.4311)/ 0.7995	7.7389E3 (16.3182)/ 0.80102	7.7620E3 (2.9283)/ 0.79861
	9	7.7852E3 (0.111484)/ 0.80293	7.7844E3 (0.4046)/ 0.80003	7.7778E3 (1.9638)/ 0.80216	7.7674E3 (11.5504)/ 0.79963	7.7774E3 (2.5778)/ 0.79894
24077	5	2.1555E4 (0.0367994)/ 0.79134	2.1555E4 (0.0515)/ 0.79144	2.1549E4 (3.8643)/ 0.79206	2.1521E4 (25.1125)/ 0.77383	2.1548E4 (2.6461)/ 0.79083
	7	2.1616E4 (0.105852)/ 0.82341	2.1616E4 (0.0533)/ 0.82342	2.1605E4 (2.7214)/ 0.81974	2.1594E4 (14.9933)/ 0.81252	2.1604E4 (3.9027)/ 0.81626
	9	2.1643E4 (0.165802)/ 0.84109	2.1643E4 (0.2071)/ 0.84107	2.1631E4 (1.6791)/ 0.83481	2.1617E4 (16.3906)/ 0.82448	2.1631E4 (2.9599)/ 0.83686
163085	5	7.9645E3 (0.0147549)/ 0.67958	7.9643E3 (0.4608)/ 0.68017	7.9597E3 (3.2964)/ 0.64757	7.9297E3 (13.4949)/ 0.65603	7.9583E3 (2.2673)/ 0.67383
	7	7.9860E3 (0.321709)/ 0.69481	7.9854E3 (0.4616)/ 0.69627	7.9776E3 (2.6945)/ 0.68766	7.9624E3 (14.9767)/ 0.67716	7.9776E3 (2.3005)/ 0.69495
	9	7.9961E3 (0.158903)/ 0.70398	7.9950E3 (0.5333)/ 0.7012	7.9870E3 (2.1345)/ 0.67902	7.9738E3 (10.6271)/ 0.68225	7.9873E3 (2.4954)/ 0.69393
101085	5	1.2968E4 (0.0213926)/ 0.76413	1.2968E4 (0.0156)/ 0.77592	1.2962E4 (3.7694)/ 0.76328	1.2943E4 (23.7702)/ 0.74733	1.2958E4 (2.1672)/ 0.76259
	7	1.3028E4 (0.406099)/ 0.79301	1.3028E4 (0.1467)/ 0.79284	1.3019E4 (4.2116)/ 0.778	1.3006E4 (17.6849)/ 0.78273	1.3010E4 (1.9442)/ 0.79271
	9	1.3057E4 (0.376722)/ 0.80922	1.3057E4 (0.3492)/ 0.80877	1.3044E4 (2.5066)/ 0.8021	1.3038E4 (12.3412)/ 0.7977	1.3041E4 (0.9811)/ 0.80401
101087	5	2.8830E4 (3.83477E - 12)/ 0.85694	2.8830E4 (0.3567)/ 0.85686	2.8827E4 (2.4413)/ 0.85724	2.8809E4 (18.3227)/ 0.84348	2.8822E4 (4.3590)/ 0.8535
	7	2.8865E4 (0.363306)/ 0.87615	2.8865E4 (0.1288)/ 0.87598	2.8858E4 (1.4591)/ 0.86972	2.8844E4 (7.8354)/ 0.84763	2.8855E4 (5.9829)/ 0.87049
	9	2.8881E4 (0.391194)/ 0.88633	2.8881E4 (0.7514)/ 0.88624	2.8874E4 (3.4566)/ 0.88024	2.8860E4 (9.4196)/ 0.86569	2.8872E4 (1.7053)/ 0.88003

(10) *Shifted Schwefel's Problem 1.2* is as follows:

$$f_{10}(x) = \sum_{i=1}^D \left(\sum_{j=1}^i z_j \right)^2 + f_{\text{bias}}, \quad z = x - o. \quad (\text{A.10})$$

(11) *Shifted Rotated High Conditioned Elliptic Function* is as follows:

$$f_{11}(x) = \sum (10^6)^{(i-1)/(D-1)} z_i^2 + f_{\text{bias}}, \quad (\text{A.11})$$

$z = (x - o) * M$, M : orthogonal matrix.

(12) *Shifted Schwefel's Problem 1.2 with Noise in Fitness* is as follows:

$$f_{12}(x) = \left(\sum_{i=1}^D \left(\sum_{j=1}^i z_j \right)^2 \right) * (1 + 0.4 |N(0, 1)|) + f_{\text{bias}}, \quad z = x - o. \quad (\text{A.12})$$

(13) *Shifted Schwefel's Problem 2.6 with Global Optimum on Bounds* is as follows:

$$f_{13}(x) = \max \{|A_i x - B_i|\} + f_{\text{bias}}, \quad i = 1, \dots, D, \quad (\text{A.13})$$

where A is a $D * D$ matrix, a_{ij} are integer random numbers in the range $[-500, 500]$, $\det(A) \neq 0$, and A_i is the i th row of A .

TABLE II: Parameters of the test functions.

f	Dimensions	Initial range	x^*	$f(x^*)$
f_1	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	0
f_2	50	$[-30, 30]^D$	$[1, 1, \dots, 1]$	0
f_3	50	$[-65.536, 65.536]^D$	$[0, 0, \dots, 0]$	0
f_4	50	$[-10, 10]^D$	$[0, 0, \dots, 0]$	0
f_5	50	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]$	0
f_6	50	$[-500, 500]^D$	$[420.9867, \dots, 420.9867]$	0
f_7	50	$[-32.768, 32.768]^D$	$[0, 0, \dots, 0]$	0
f_8	50	$[-600, 600]^D$	$[0, 0, \dots, 0]$	0
f_9	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
f_{10}	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
f_{11}	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
f_{12}	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
f_{13}	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-310
f_{14}	50	$[-100, 100]^D$	$[0, 0, \dots, 0]$	390
f_{15}	50	No bounds	$[0, 0, \dots, 0]$	-180
f_{16}	50	$[-32, 32]^D$	$[0, 0, \dots, 0]$	-140
f_{17}	50	$[-5, 5]^D$	$[0, 0, \dots, 0]$	-330
f_{18}	50	$[-5, 5]^D$	$[0, 0, \dots, 0]$	-330
f_{19}	50	$[-0.5, 0.5]^D$	$[0, 0, \dots, 0]$	-460
f_{20}	50	$[-\pi, \pi]^D$	$[\alpha_1, \alpha_2, \dots, \alpha_D]$	-460

$B_i = A_i * o$, o is a $D * 1$ vector, and o_i are random number in the range $[-100, 100]$.

(14) *Shifted Rosenbrock's Function* is as follows:

$$f_{14}(x) = \sum_{i=1}^{D-1} \left(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2 \right) + f_{\text{bias}}. \quad (\text{A.14})$$

(15) *Shifted Rotated Griewank's Function without Bounds* is as follows:

$$f_{15}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}. \quad (\text{A.15})$$

(16) *Shifted Rotated Ackley's Function with Global Optimum on Bounds* is as follows:

$$f_{16}(x) = -\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D z_i^2}\right) - \exp\left(\sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e + f_{\text{bias}}, \quad z = (x - o) * M. \quad (\text{A.16})$$

(17) *Shifted Rastrigin's Function* is as follows:

$$f_{17}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}}, \quad (\text{A.17})$$

$$z = x - o.$$

(18) *Shifted Rotated Rastrigin's Function* is as follows:

$$f_{18}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}}, \quad (\text{A.18})$$

$$z = (x - o) * M.$$

(19) *Shifted Rotated Weierstrass Function* is as follows:

$$f_{19}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - D \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k * 0.5)] + f_{\text{bias}_{11}},$$

$$a = 0.5, \quad b = 3, \quad k_{\max} = 20, \quad z = (x - o) * M. \quad (\text{A.19})$$

(20) *Schwefel's Problem 2.13* is as follows:

$$f_{20}(x) = \sum_{i=1}^D (A_i - B_i(x))^2 + f_{\text{bias}_{12}},$$

$$A_i = \sum_{j=1}^D (a_{ij} \sin a_j + b_{ij} \cos a_j), \quad (\text{A.20})$$

$$B_i = \sum_{j=1}^D (a_{ij} \sin x_j + b_{ij} \cos x_j),$$

where A, B are two $D * D$ matrix, a_{ij}, b_{ij} are integer random numbers in the range $[-100, 100]$, and a_j are random numbers in the range $[-\pi, \pi]$.

A.3. *Parameters of the Test Functions.* The dimensions, initialization ranges, global optima x^* , and the corresponding fitness value $f(x^*)$ of each function are listed in Table II.

Conflict of Interests

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

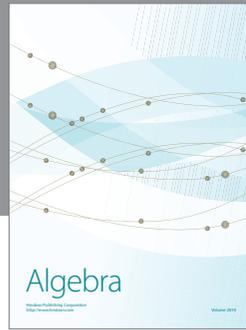
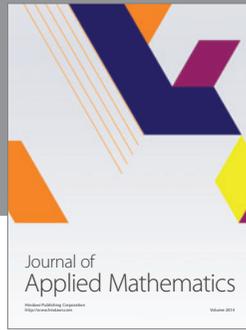
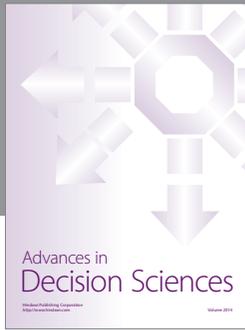
Acknowledgments

This research is partially supported by National Natural Science Foundation of China under Grant nos. 71001072 and 71271140 and the National High Technology Research and Development Program of China (863 Program) (no. 2014AA052101-3).

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] 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, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.
- [4] D. Tsai, "A fast thresholding selection procedure for multimodal and unimodal histograms," *Pattern Recognition Letters*, vol. 16, no. 6, pp. 653–666, 1995.
- [5] A. D. Brink, "Minimum spatial entropy threshold selection," *IEE Proceedings: Vision, Image and Signal Processing*, vol. 142, no. 3, pp. 128–132, 1995.
- [6] H. D. Cheng, J. Chen, and J. Li, "Threshold selection based on fuzzy c-partition entropy approach," *Pattern Recognition*, vol. 31, no. 7, pp. 857–870, 1998.
- [7] 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.
- [8] 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.
- [9] 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.
- [10] 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.
- [11] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Applied Soft Computing Journal*, vol. 13, no. 6, pp. 3066–3091, 2013.
- [12] C. Ozturk, E. Hancer, and D. Karaboga, "Improved clustering criterion for image clustering with artificial bee colony algorithm," *Pattern Analysis and Applications*, 2014.
- [13] S. Sarkar and S. Das, "Multilevel image thresholding based on 2D histogram and maximum Tsallis entropy—a differential evolution approach," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4788–4797, 2013.
- [14] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Computer Vision—ACCV 2009*, vol. 5994 of *Lecture Notes in Computer Science*, pp. 135–146, Springer, Berlin, Germany, 2010.
- [15] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Unsupervised segmentation of natural images via lossy data compression," *Computer Vision and Image Understanding*, vol. 110, no. 2, pp. 212–225, 2008.
- [16] D. Karaboga and B. Akay, "A survey: algorithms simulating bee swarm intelligence," *Artificial Intelligence Review*, vol. 31, no. 1–4, pp. 61–85, 2009.
- [17] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.
- [18] F. Liu and Y. Wang, "The reversible data hiding with lower distortion based on histogram shifting," *Information and Control*, vol. 42, no. 5, pp. 595–600, 2013.
- [19] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
- [20] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [21] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing: 12th International Fuzzy Systems Association World Congress, IFSA 2007, Cancun, Mexico, June 18–21, 2007*, P. Melin, O. Castillo, L. T. Aguilar, J. Kacprzyk, and W. Pedrycz, Eds., vol. 4529 of *Lecture Notes in Computer Science*, pp. 789–798, Springer, Berlin, Germany, 2007.
- [22] B. Basturk and D. Karaboga, "An Artificial Bee Colony (ABC) algorithm for numerical function optimization," in *Proceedings of the IEEE, Swarm Intelligence Symposium*, Indianapolis, Ind, USA, 2006.
- [23] 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.
- [24] M. A. Potter and K. A. de Jong, "Cooperative coevolution: an architecture for evolving coadapted subcomponents," *Evolutionary Computation*, vol. 8, no. 1, pp. 1–29, 2000.
- [25] Z. Yang, K. Tang, and X. Yao, "Differential evolution for high-dimensional function optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 3523–3530, Singapore, September 2007.
- [26] Y. Shi, H. Teng, and Z. Li, "Cooperative co-evolutionary differential evolution for function optimization," in *Proceedings of the 1st International Conference on Natural Computation (ICNC '05)*, pp. 1080–1088, August 2005.
- [27] 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.
- [28] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [29] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.
- [30] H. N. Chen, Y. L. Zhu, K. Hu, and X. He, "Hierarchical swarm model: a new approach to optimization," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 379649, 30 pages, 2010.

- [31] B. Niu, Y. Zhu, X. He, and H. Wu, "MCPSO: a multi-swarm cooperative particle swarm optimizer," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1050–1062, 2007.
- [32] S. Janson and M. Middendorf, "A hierarchical particle swarm optimizer and its adaptive variant," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 6, pp. 1272–1282, 2005.
- [33] H. Chen, Y. Zhu, K. Hu, and X. He, "Hierarchical swarm model: a new approach to optimization," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 379649, 30 pages, 2010.
- [34] Y. Peng and B. Lu, "A hierarchical particle swarm optimizer with latin sampling based memetic algorithm for numerical optimization," *Applied Soft Computing*, vol. 13, no. 5, pp. 2823–2836, 2013.
- [35] J. Kennedy and R. Mendes, "Topological structure and particle swarm performance," in *Proceedings of the 4th Congress on Evolutionary Computation (CEC '02)*, pp. 1671–1676, 2002.
- [36] 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.
- [37] B. Y. Qu, P. N. Suganthan, and J. J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 601–614, 2012.
- [38] B. Y. Qu, P. N. Suganthan, and S. Das, "A distance-based locally informed particle swarm model for multimodal optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 387–402, 2013.
- [39] S. Sumathi, T. Hamsapriya, and P. Surekha, *Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab*, Springer, New York, NY, USA, 2008.
- [40] D. H. Wolpert and W. G. Macready, "No free lunch theorems for search," Santa Fe Institute SFI-TR-95-02-010, 1995.
- [41] W. Zou, Y. Zhu, H. Chen, and X. Sui, "A clustering approach using cooperative artificial bee colony algorithm," *Discrete Dynamics in Nature and Society*, vol. 2010, Article ID 459796, 16 pages, 2010.
- [42] 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.
- [43] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [44] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep. #2005005, Nanyang Technological University, Singapore; IIT, KanGAL Report, Kanpur, India, 2005.
- [45] 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.
- [46] R. Unnikrishnan and M. Hebert, "Measures of similarity," in *Proceedings of the 7th IEEE Workshop on Applications of Computer Vision (WACV '05)*, pp. 394–400, January 2005.
- [47] 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.
- [48] P. Yin, "Multilevel minimum cross entropy threshold selection based on particle swarm optimization," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 503–513, 2007.
- [49] L. Cao, P. Bao, and Z. Shi, "The strongest schema learning GA and its application to multilevel thresholding," *Image and Vision Computing*, vol. 26, no. 5, pp. 716–724, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

