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 Song 1

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
minimaums and loss of population diversity [20].
Comparing with other evolutionary and swarm intel-
ligence 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 optimi-
zation 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 group-
ing is incorporated in this hierarchical framework,
which can decompose the complex high-dimensional vec-
tors into several smaller part-dimensional com-
ponents 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 popu-
lation 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 seg-
mentation 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:

\[ x_{i,j} = x_{j}^{\text{min}} + \text{rand} \times (0,1) \left( x_{j}^{\text{max}} - x_{j}^{\text{min}} \right), \]

where \( i = 1, 2, \ldots, SN \), \( j = 1, 2, \ldots, D \). \( SN \) is the number of
food sources. \( D \) is the number of variables, that is, problem
dimension. \( x_{j}^{\text{min}} \) and \( x_{j}^{\text{max}} \) 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 gen-
erated from the old food source of each employed bee in its
memory using the following expression:

\[ v_{i,j} = x_{i,j} + \varphi \left( x_{i,j} - x_{k,j} \right), \]

where \( k \) is a randomly selected food source and must be
different from \( i \); \( j \) is a randomly chosen indexes; \( \varphi \) 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:

\[ P_{i} = \frac{\text{fitness}_{i}}{\sum_{j=1}^{SN} \text{fitness}_{j}}, \]

where \( \text{fitness}_{i} \) is the fitness value of \( i \)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 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 subpopu-
lation 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

}\]
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 $D = K \times s$). The $j$th subpopulation is denoted as $P_j$, $j \in [1 \cdots K]$.

Step 2. Construct complete evolving solution $G_{best}$, which is the concatenation of the best subcomponents’ solutions $P_j$ by fowling:

$$G_{best} = (P_1 \cdot g, P_2 \cdot g, P_j \cdot g \cdots P_K \cdot g),$$

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 $G_{best}$ by using the $i$th component of individual $X_i$. Calculate the new solution fitness: $f(\text{new}G_{best}(P_1 \cdot g, P_2 \cdot g, X_i, \ldots, P_K \cdot g))$. If $f(\text{new}G_{best}) < f(G_{best})$, then $G_{best}$ is replaced by $\text{new}G_{best}$.

(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 $G_{best}$ 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

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

$$K \subset G.$$ (5)

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 $P_i$ 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 $P_i$’s neighborhood (i.e., ring topology) with higher fitness have larger probability to be
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]:

$$s_{\text{new}} = \text{rand}(0,1) \times \text{parent}1 + \text{rand}(0,1) \times \text{parent}2,$$

where $s_{\text{new}}$ 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 $S \times CR$ ($CR$ 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., $S \times CR$ individuals), a medium level of individuals and the worst individuals. To maintain the population diversity, we randomly select one replacing approach at each 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 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.

#### Table 1: Parameters of the HABC.

<table>
<thead>
<tr>
<th>HABC = $(N, M, P_{ij}, C, CR, T, O, S)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
</tr>
<tr>
<td>K</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>i</td>
</tr>
<tr>
<td>j</td>
</tr>
<tr>
<td>$P_{ij}$</td>
</tr>
<tr>
<td>CR</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>O</td>
</tr>
</tbody>
</table>
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 \( G_best = (P_{i1} \cdot g, P_{i2} \cdot g, \ldots, P_{il-1} \cdot g, z, P_{il+1} \cdot g, \ldots P_{iK} \cdot g) \),
which consists of the \( S \)-dimensions best solution \( P_{il} \).
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:
\[ newGbest = (P_{ih} \cdot g, P_{i2} \cdot g, \ldots, P_{ij} \cdot x_k, \ldots, P_{iK} \cdot g) \]
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
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
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 \((f_1–f_5)\) with different CRs, it is visible that HABC converged faster than all other algorithms. HABC was able to consistently find the minimum to functions \( f_1, f_2, \text{ and } f_3 \) 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 CABC also can consistently find the minimum of \( f_5 \) within relatively more FEs while other algorithms perform poorer. This can be

<table>
<thead>
<tr>
<th>Swarm number ((N))</th>
<th>2</th>
<th>6</th>
<th>10</th>
<th>14</th>
<th>18</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 ) Mean</td>
<td>0.3e+004</td>
<td>0.1e+004</td>
<td>0.1e+004</td>
<td>0.1e+004</td>
<td>0.9e+003</td>
</tr>
<tr>
<td>( f_1 ) Std.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_2 ) Mean</td>
<td>0.1e+004</td>
<td>0.1e+004</td>
<td>0.9e+003</td>
<td>0.8e+003</td>
<td>0.8e+003</td>
</tr>
<tr>
<td>( f_2 ) Std.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_3 ) Mean</td>
<td>0.5e+004</td>
<td>0.2e+004</td>
<td>0.1e+004</td>
<td>0.1e+004</td>
<td>0.1e+004</td>
</tr>
<tr>
<td>( f_3 ) Std.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_9 ) Mean</td>
<td>0.4e+004</td>
<td>0.3e+004</td>
<td>0.2e+004</td>
<td>0.2e+004</td>
<td>0.2e+004</td>
</tr>
<tr>
<td>( f_9 ) Std.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_{10} ) Mean</td>
<td>6.4e+002</td>
<td>2.3e+002</td>
<td>1.3e+001</td>
<td>1.3e+001</td>
<td>1.2e+001</td>
</tr>
<tr>
<td>( f_{10} ) Std.</td>
<td>5.9e+001</td>
<td>4.9e+001</td>
<td>3.6e+001</td>
<td>2.5e+001</td>
<td>1.2e+001</td>
</tr>
<tr>
<td>( f_{11} ) Mean</td>
<td>5.8e+003</td>
<td>3.1e+003</td>
<td>3.1e+003</td>
<td>3.1e+003</td>
<td>3.1e+003</td>
</tr>
<tr>
<td>( f_{11} ) Std.</td>
<td>2.2e+002</td>
<td>1.0e+002</td>
<td>7.0e+001</td>
<td>5.1e+001</td>
<td>3.6e+001</td>
</tr>
<tr>
<td>( f_{13} ) Mean</td>
<td>3.3e+004</td>
<td>1.9e+004</td>
<td>9.3e+003</td>
<td>9.3e+003</td>
<td>8.9e+003</td>
</tr>
<tr>
<td>( f_{13} ) Std.</td>
<td>6.9e+01</td>
<td>3.6e+01</td>
<td>4.1e+03</td>
<td>9.6e+02</td>
<td>7.0e+02</td>
</tr>
</tbody>
</table>

### Table 3: Results of HABC on benchmark functions \((f_1–f_5, f_8–f_{12})\) with different CRs. In bold are the best results.

<table>
<thead>
<tr>
<th>Function</th>
<th>Mean</th>
<th>Std.</th>
<th>Mean</th>
<th>Std.</th>
<th>Mean</th>
<th>Std.</th>
<th>Mean</th>
<th>Std.</th>
<th>Mean</th>
<th>Std.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>2.34e−109</td>
<td>5.65e−10</td>
<td>5.32e−90</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td></td>
</tr>
<tr>
<td>( f_2 )</td>
<td>3.97e−004</td>
<td>3.12e−006</td>
<td>2.65e−005</td>
<td>6.87</td>
<td>3.12e−006</td>
<td>3.54e−007</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_3 )</td>
<td>2.67e−101</td>
<td>2.21e−194</td>
<td>1.67e−267</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_9 )</td>
<td>6.09e−067</td>
<td>3.59e−140</td>
<td>1.13e−193</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td>0.050000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{10} )</td>
<td>4.70e+001</td>
<td>5.97e+001</td>
<td>3.59e+002</td>
<td>2.89e+003</td>
<td>8.76e+001</td>
<td>9.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{11} )</td>
<td>3.11e−120</td>
<td>8.14e−091</td>
<td>5.43e−103</td>
<td>7.53e−073</td>
<td>3.19e−099</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{12} )</td>
<td>2.01e+002</td>
<td>5.83e+002</td>
<td>7.34e+003</td>
<td>8.99e+003</td>
<td>2.45e+003</td>
<td>6.43e+002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f_{13} )</td>
<td>2.14e−098</td>
<td>5.93e−131</td>
<td>2.34e−086</td>
<td>4.31e−096</td>
<td>5.17e−134</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Discrete Dynamics in Nature and Society
**Table 4: Performance of HABC on 50-D \( f_1-f_5 \ &f_9-f_{13} \) with the different grouping number \( K \). In bold are the best results.**

<table>
<thead>
<tr>
<th>Function</th>
<th>( K \subset S )</th>
<th>( K = 5 )</th>
<th>( K = 10 )</th>
<th>( K = 25 )</th>
<th>( K = 50 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>Mean 0 (95000)</td>
<td>3.01e-080</td>
<td>4.43e-043</td>
<td>7.32e-042</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std. 0</td>
<td>2.23e-001</td>
<td>5.65e-043</td>
<td>2.43e-041</td>
<td></td>
</tr>
<tr>
<td>( f_2 )</td>
<td>Mean 1.87e-014</td>
<td>1.58</td>
<td>1.87</td>
<td>6.67e-001</td>
<td>1.56e-022</td>
</tr>
<tr>
<td></td>
<td>Std. 1.65e-014</td>
<td>3.48e-001</td>
<td>2.38e-001</td>
<td>1.23</td>
<td>1.21e-022</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>Mean 0</td>
<td>4.34e-014</td>
<td>6.87e-013</td>
<td>9.54e-013</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std. 0</td>
<td>1.51e-013</td>
<td>3.53e-013</td>
<td>2.68e-013</td>
<td></td>
</tr>
<tr>
<td>( f_4 )</td>
<td>Mean 0 (177000)</td>
<td>3.03e-251</td>
<td>2.33e-135</td>
<td>7.82e-080</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std. 0</td>
<td>8.03e-301</td>
<td>3.78e-150</td>
<td>1.10e-103</td>
<td></td>
</tr>
<tr>
<td>( f_10 )</td>
<td>Mean 7.15e + 001</td>
<td>2.78e + 001</td>
<td>1.58e + 001</td>
<td>8.74</td>
<td>5.61</td>
</tr>
<tr>
<td></td>
<td>Std. 2.33e + 001</td>
<td>1.03e +001</td>
<td>5.48</td>
<td>8.65e-001</td>
<td>5.91</td>
</tr>
<tr>
<td>( f_11 )</td>
<td>Mean 0</td>
<td>6.93e - 210</td>
<td>4.19e -110</td>
<td>1.03e -093</td>
<td>8.56e -081</td>
</tr>
<tr>
<td></td>
<td>Std. 0</td>
<td>3.33e - 310</td>
<td>3.93e -170</td>
<td>9.51e -104</td>
<td>5.61e -094</td>
</tr>
<tr>
<td>( f_12 )</td>
<td>Mean 7.57e + 002</td>
<td>2.01e +003</td>
<td>4.78e + 003</td>
<td>7.98e + 003</td>
<td>1.13e + 004</td>
</tr>
<tr>
<td></td>
<td>Std. 5.31e + 002</td>
<td>8.10e +002</td>
<td>1.01 e +003</td>
<td>2.88e + 003</td>
<td>7.03e + 003</td>
</tr>
<tr>
<td>( f_13 )</td>
<td>Mean 0</td>
<td>5.34e - 110</td>
<td>1.44e -088</td>
<td>5.46e -054</td>
<td>9.51e -027</td>
</tr>
<tr>
<td></td>
<td>Std. 0</td>
<td>3.75e - 067</td>
<td>1.39e -045</td>
<td>3.91e -063</td>
<td>1.93e -010</td>
</tr>
</tbody>
</table>

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 \( K \subset S \), in comparison with CABC, CPSO, CMA-ES, ABC, PSO, and GA algorithms. From the experimental results in terms of mean 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 \( N \times K \times M \times Cost_a + N \times Cost_c \). 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
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

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

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

Maximize $f(t) = w_0 w_1 (u_0 - u_1)^2$, \hspace{1cm} (8)

where

\[
w_0 = \sum_{i=0}^{t-1} P(i), \quad u_0 = \sum_{i=0}^{t-1} i \times P(i)/w_0,
\]

\[
w_1 = \sum_{i=t}^{L-1} P(i), \quad u_1 = \sum_{i=t}^{L-1} i \times P(i)/w_1,
\]

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

\[
f(t) = \delta^2 - w_0 \delta^2 - w_1 \delta^2,
\]

\[
\delta = u_0 - u_1.
\]
where $w_0$, $w_1$, $u_0$, and $u_1$ are the same as given in (9), and

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

$$
\delta_1 = \sum_{i=0}^{L-1} \frac{(i-u_1)^2 \times P_i}{w_1},
$$

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

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

Figure 5: ANOVA results of all algorithms. Here, 1 to 7 are index of HABC, CABC, CPSO, CMA-ES, ABC, PSO, and GA, respectively.
Table 5: Performance of all algorithms on basic benchmark functions $f_1$–$f_8$.

<table>
<thead>
<tr>
<th>Function</th>
<th>HABC</th>
<th>CABC</th>
<th>CPSO</th>
<th>CMA-ES</th>
<th>ABC</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
<td>Mean</td>
</tr>
<tr>
<td>$f_1$ Std</td>
<td>1.01e-137</td>
<td>3.01e-014</td>
<td>0</td>
<td>9.94e-016</td>
<td>1.85e-009</td>
<td>4.03e-001</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>$f_2$ Std</td>
<td>1.741e-3</td>
<td>2.45e-137</td>
<td>0</td>
<td>7.85e-016</td>
<td>1.01e-009</td>
<td>1.05e-001</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>$f_3$ Std</td>
<td>3.89e + 001</td>
<td>2.66e + 001</td>
<td>0</td>
<td>3.33e + 001</td>
<td>1.99e + 003</td>
<td>7.45e + 001</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>$f_4$ Std</td>
<td>7.31e + 001</td>
<td>4.71e + 001</td>
<td>0</td>
<td>3.97</td>
<td>3.15e + 003</td>
<td>3.44e + 001</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>$f_5$ Std</td>
<td>3.02e + 032</td>
<td>2.15e + 032</td>
<td>4.97e - 009</td>
<td>2.34e + 032</td>
<td>8.07e - 014</td>
<td>8.27e - 001</td>
<td>9.10e - 001</td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$f_6$ Std</td>
<td>4.31e + 002</td>
<td>1.96</td>
<td>4.31e + 002</td>
<td>1.84e - 001</td>
<td>2.48e + 001</td>
<td>8.01e + 002</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>$f_7$ Std</td>
<td>8.01e + 016</td>
<td>3.10e + 002</td>
<td>2.32e + 003</td>
<td>5.03e + 003</td>
<td>1.32e + 001</td>
<td>5.14e + 003</td>
<td>2.79e + 002</td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$f_8$ Std</td>
<td>3.76e + 032</td>
<td>2.08e - 008</td>
<td>8.75e - 034</td>
<td>5.03e - 014</td>
<td>5.01e - 001</td>
<td>5.11e - 001</td>
<td></td>
</tr>
<tr>
<td>Rank</td>
<td>3</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

Total rank 1 6 16 18 5 16 22 27

Total rank 2 4 9 18 22 15 23 25

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 $I(I_i^k = 1)$ 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]

$$PRI(S_{test} \vert S_k) = \frac{1}{\binom{N}{2}} \sum_{i,j} \left[ \sum_{k} \left[ I(I_i^k = 1) \right] p_{ij} + I(I_i^k \neq 1) \right].$$

(13)

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

$$c_{ij} = I(I_i^k = 1).$$

(14)
Table 6: Performance of all algorithms on CEC2005 functions $f_9$–$f_{20}$.

<table>
<thead>
<tr>
<th>Function</th>
<th>HABC</th>
<th>CABC</th>
<th>CPSO</th>
<th>CMA-ES</th>
<th>ABC</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_9$</td>
<td>Mean</td>
<td>2.85e−013</td>
<td>3.57e+002</td>
<td>4.76e−014</td>
<td>3.65e−012</td>
<td>1.88e+001</td>
<td>7.05e+001</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$f_{10}$</td>
<td>Mean</td>
<td>2.34e+001</td>
<td>1.67e+003</td>
<td>4.86e+003</td>
<td>0</td>
<td>4.64e+003</td>
<td>2.14e+002</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>$f_{11}$</td>
<td>Mean</td>
<td>4.04e−013</td>
<td>4.25e+002</td>
<td>3.24e−014</td>
<td>1.69e−11</td>
<td>3.55e+001</td>
<td>3.91e+001</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>$f_{12}$</td>
<td>Mean</td>
<td>4.07e+003</td>
<td>2.43e+004</td>
<td>7.31e−014</td>
<td>4.16e+003</td>
<td>1.01e+002</td>
<td>2.50e+003</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>$f_{13}$</td>
<td>Mean</td>
<td>2.07e+001</td>
<td>3.13e+001</td>
<td>0</td>
<td>1.85e+001</td>
<td>1.20e−003</td>
<td>1.05e+002</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$f_{14}$</td>
<td>Mean</td>
<td>5.48</td>
<td>1.39e+006</td>
<td>5.13</td>
<td>7.40</td>
<td>3.27e+001</td>
<td>1.23e+004</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>$f_{15}$</td>
<td>Mean</td>
<td>1.16e+003</td>
<td>9.31e+002</td>
<td>1.84e+003</td>
<td>1.37e+003</td>
<td>3.59e+003</td>
<td>1.46e+003</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$f_{16}$</td>
<td>Mean</td>
<td>2.91e+001</td>
<td>3.86e+002</td>
<td>3.37e+001</td>
<td>1.51e+001</td>
<td>3.86e+001</td>
<td>4.47e+001</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>$f_{17}$</td>
<td>Mean</td>
<td>1.65e+003</td>
<td>1.70e+002</td>
<td>2.70e+001</td>
<td>1.67e−002</td>
<td>5.27e+001</td>
<td>3.36e+001</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>$f_{18}$</td>
<td>Mean</td>
<td>2.68e+002</td>
<td>2.33e+002</td>
<td>5.42e+002</td>
<td>1.20e+002</td>
<td>4.49e+002</td>
<td>2.21e+002</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>$f_{19}$</td>
<td>Mean</td>
<td>1.01e+002</td>
<td>4.57e+002</td>
<td>3.41e+001</td>
<td>3.71e+002</td>
<td>7.16e+002</td>
<td>1.11e+003</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>$f_{20}$</td>
<td>Mean</td>
<td>5.84e+002</td>
<td>3.11e+003</td>
<td>1.47e+003</td>
<td>9.41e+002</td>
<td>4.54e+003</td>
<td>8.01e+004</td>
</tr>
<tr>
<td></td>
<td>Rank</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>6</td>
</tr>
</tbody>
</table>

Then, the PRI can be written as

$$\text{PR}(S_{\text{test}} | S_k) = \frac{1}{N} \sum_{i,j \leq k} \left[ c_{ij} p_{ij} + (1 - c_{ij}) (1 - p_{ij}) \right]. \quad (15)$$

This measure takes values in [0, 1], where 0 means $S_{\text{test}}$ and $S_k$ have no similarities (i.e., when $S$ consists of a single cluster and each segmentation in $\{S_1, S_2, \ldots, S_k\}$ 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/bsds/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
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, \( K \subset \{1, 2\} \); for 3-dimensional problem, \( K \subset \{1, 3\} \); for 4-dimensional case, \( K \subset \{1, 2, 4\} \); for 5-dimensional case, \( K \subset \{1, 5\} \); for 7-dimensional case, \( K \subset \{1, 7\} \); for 9-dimensional case, \( K \subset \{1, 3, 9\} \). 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 correlation 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
Figure 7: Test images and their histograms.
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:

\[ f_1(x) = \sum_{i=1}^{n} x_i^2. \] (A.1)

(2) Rosenbrock Function is as follows:

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

(3) Quadric Function is as follows:

\[ f_3(x) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n} x_{i,j}^2 \right). \] (A.3)

(4) Sin Function is as follows:

\[
\begin{align*}
    f_4(x) &= \frac{\pi}{n} \left\{ 10 \sin^2 \pi x_1 \\
    &+ \sum_{i=1}^{n-1} (x_i - 1)^2 \left( 1 + 10 \sin^2 \pi x_{i+1} \right) + (x_n - 1)^2 \right\}.
\end{align*}
\] (A.4)

(5) Rastrigin Function is as follows:

\[ f_5(x) = \sum_{i=1}^{n} \left( x_i^2 - 10 \cos (2\pi x_i) \right) + 10. \] (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.

<table>
<thead>
<tr>
<th>Image</th>
<th>Objective values</th>
<th>Optimal thresholds</th>
<th>PRI</th>
<th>Objective values</th>
<th>Optimal thresholds</th>
<th>PRI</th>
<th>Objective values</th>
<th>Optimal thresholds</th>
<th>PRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>38092</td>
<td>2.5064E4</td>
<td>101,180</td>
<td>0.74248</td>
<td>2.5258E4</td>
<td>77,132,190</td>
<td>0.78598</td>
<td>2.5345.4</td>
<td>64,111,152,197</td>
<td>0.7994</td>
</tr>
<tr>
<td>14037</td>
<td>7.5361E3</td>
<td>61,128</td>
<td>0.73614</td>
<td>7.6593E3</td>
<td>33,773,131</td>
<td>0.8019</td>
<td>7.7151E4</td>
<td>33,73,109,147</td>
<td>0.79968</td>
</tr>
<tr>
<td>24077</td>
<td>2.1133E4</td>
<td>81,175</td>
<td>0.6821</td>
<td>2.1411E4</td>
<td>66,126,199</td>
<td>0.73976</td>
<td>2.1499E4</td>
<td>63,112,163,217</td>
<td>0.7637</td>
</tr>
<tr>
<td>163085</td>
<td>7.8365E3</td>
<td>69,110</td>
<td>0.61394</td>
<td>7.9063E3</td>
<td>57,85,119</td>
<td>0.65081</td>
<td>7.9426E3</td>
<td>52,77,102,132</td>
<td>0.6706</td>
</tr>
<tr>
<td>101085</td>
<td>1.2559E4</td>
<td>87,174</td>
<td>0.63896</td>
<td>1.2579E4</td>
<td>58,113,184</td>
<td>0.70822</td>
<td>1.2904E4</td>
<td>47,90,140,200</td>
<td>0.73810</td>
</tr>
<tr>
<td>101087</td>
<td>2.855E4</td>
<td>107,196</td>
<td>0.76849</td>
<td>2.8714E4</td>
<td>89,135,206</td>
<td>0.81706</td>
<td>2.8791E4</td>
<td>67,105,144,209</td>
<td>0.84213</td>
</tr>
</tbody>
</table>

Mean CPU time: 1.4427, 61.421, 2644.543

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

<table>
<thead>
<tr>
<th>Dim.</th>
<th>Alg.</th>
<th>HABC</th>
<th>ABC</th>
<th>PSO</th>
<th>CMA-ES</th>
<th>EGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td></td>
<td>0.8575</td>
<td>0.7954</td>
<td>0.2126</td>
<td>0.2438</td>
<td>2.4755</td>
</tr>
<tr>
<td>3D</td>
<td></td>
<td>0.8956</td>
<td>0.8422</td>
<td>0.2398</td>
<td>0.2876</td>
<td>3.1928</td>
</tr>
<tr>
<td>4D</td>
<td></td>
<td>0.8982</td>
<td>0.8607</td>
<td>0.2395</td>
<td>0.306</td>
<td>4.3953</td>
</tr>
<tr>
<td>5D</td>
<td></td>
<td>0.6322</td>
<td>0.8861</td>
<td>0.2677</td>
<td>0.3061</td>
<td>1.1138</td>
</tr>
<tr>
<td>7D</td>
<td></td>
<td>1.0136</td>
<td>0.9830</td>
<td>0.2944</td>
<td>0.4403</td>
<td>1.6599</td>
</tr>
<tr>
<td>9D</td>
<td></td>
<td>1.0461</td>
<td>0.9201</td>
<td>0.4356</td>
<td>0.4445</td>
<td>2.3235</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Image M − 1</th>
<th>HABC (Objective values (standard deviation))/PRI</th>
<th>ABC</th>
<th>PSO</th>
<th>CMA-ES</th>
<th>EGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>38092</td>
<td>2.5064E4 (3.83477E − 12)/0.74248</td>
<td>2.5064E4 (3.8348E − 12)/0.74248</td>
<td>2.5064E4 (3.8348E − 12)/0.74248</td>
<td>2.5047E4 (18.3953)/0.73305</td>
<td>2.5063E4 (0.2663)/0.74268</td>
</tr>
<tr>
<td>14037</td>
<td>2.5259E4 (0.025736)/0.78598</td>
<td>2.5259E4 (0.0259)/0.78611</td>
<td>2.5259E4 (0.0462)/0.78546</td>
<td>2.5235E4 (39.6597)/0.77949</td>
<td>2.5256E4 (1.4455)/0.78657</td>
</tr>
<tr>
<td>24077</td>
<td>2.3435E4 (0.0192)/0.7994</td>
<td>2.3435E4 (0.8140)/0.79932</td>
<td>2.3435E4 (0.8140)/0.80059</td>
<td>2.5276E4 (65.1685)/0.77711</td>
<td>2.5340E4 (2.0817)/0.79761</td>
</tr>
<tr>
<td>163085</td>
<td>7.5360E3 (0.00682035)/0.73621</td>
<td>7.5360E3 (0.0068)/0.73621</td>
<td>7.5360E3 (0.0068)/0.73614</td>
<td>7.4730E3 (99.7498)/0.73234</td>
<td>7.5359E3 (0.2176)/0.73693</td>
</tr>
<tr>
<td>101085</td>
<td>7.6593E3 (0.58692E − 13)/0.8019</td>
<td>7.6593E3 (0.58692E − 13)/0.8019</td>
<td>7.6593E3 (0.2069)/0.80121</td>
<td>7.6006E3 (47.7619)/0.76189</td>
<td>7.6567E3 (1.4649)/0.79826</td>
</tr>
<tr>
<td>101087</td>
<td>7.7151E3 (0.58692E − 13)/0.79968</td>
<td>7.7151E3 (0.58692E − 13)/0.79968</td>
<td>7.7151E3 (0.19174E − 12)/0.79803</td>
<td>7.6729E3 (57.4010)/0.78318</td>
<td>7.7110E3 (2.3737)/0.79919</td>
</tr>
</tbody>
</table>

(6) Schwefel Function is as follows:

\[ f_6(x) = D \ast 418.9829 + \sum_{i=1}^{D} - x_i \sin \left( \sqrt[n]{|x_i|} \right). \]  \hspace{1cm} (A.6)

(8) Griewank Function is as follows:

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

(7) Ackley's Function is as follows:

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

(9) Shifted Sphere Function is as follows:

\[ f_9(x) = \sum_{i=1}^{n} z_i^2 + f_{bias1}, \quad z = x - o. \]  \hspace{1cm} (A.9)

A.2. CEC2005 Function

Schwefel Function is as follows:

\[ f(x) = \sum_{i=1}^{D} z_i^2 + f_{bias1}, \quad z = x - o. \]  \hspace{1cm} (A.9)
Discrete Dynamics in Nature and Society

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

\[
f_{10}(x) = \sum_{j=1}^{D} \left( \sum_{i=1}^{j} z_{ij} \right)^{2} + f_{\text{bias}}, \quad z = x - o.
\]

(A.10)

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

\[
f_{11}(x) = \sum \left( 10^{6} \right)^{\left( D-1 \right) / D} z_{ij}^{2} + f_{\text{bias}}, \quad z = (x - o) \ast M, \quad M : \text{orthogonal matrix}.
\]

(A.11)

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

\[
f_{12}(x) = \left( \sum_{j=1}^{D} \left( \sum_{i=1}^{j} z_{ij} \right)^{2} \right) \ast \left( 1 + 0.4 |N(0,1)| \right) + f_{\text{bias}}, \quad z = x - o.
\]

(A.12)

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

\[
f_{13}(x) = \max \left( |A_{i} x - B_{i}| + f_{\text{bias}}, \quad i = 1, \ldots, D \right), \quad A_{i} \ast M : \text{integer random numbers in the range} [-500,500], \det(A) \neq 0, \quad A_{i} \text{ is the } i\text{ th row of } A.
\]

(A.13)
Table II: Parameters of the test functions.

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

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

(14) Shifted Rosenbrock’s Function is as follows:
\[
f_{14}(x) = \sum_{i=1}^{D-1} 100(z_i^2 - z_{i+1})^2 + (z_D^2 - 1)^2 + f_{bias} \tag{A.14}
\]

(15) Shifted Rotated Griewank’s Function without Bounds is as follows:
\[
f_{15}(x) = \sum_{i=1}^{D} \frac{z_i^2}{4000} + \prod_{i=1}^{D} \cos \left( \frac{z_i}{\sqrt{i}} \right) + 1 + f_{bias} \tag{A.15}
\]

(16) Shifted Rotated Ackley’s Function with Global Optimum on Bounds is as follows:
\[
f_{16}(x) = -\exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} z_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^{D} \cos (2\pi z_i) \right) + 20 + \epsilon + f_{bias}, \quad z = (x - o) \ast M. \tag{A.16}
\]

(17) Shifted Rastrigin’s Function is as follows:
\[
f_{17}(x) = \sum_{i=1}^{D} (z_i^2 - 10 \cos (2\pi z_i) + 10) + f_{bias}, \tag{A.17}
\]

\[z = x - o.\]

(18) Shifted Rotated Rastrigin’s Function is as follows:
\[
f_{18}(x) = \sum_{i=1}^{D} (z_i^2 - 10 \cos (2\pi z_i) + 10) + f_{bias}, \tag{A.18}
\]

\[z = (x - o) \ast M.\]

(19) Shifted Rotated Weierstrass Function is as follows:
\[
f_{19}(x) = \sum_{j=1}^{D} \sum_{k=0}^{k_{max}} \left[ a^k \cos \left( 2\pi b^k (z_j + 0.5) \right) \right] - D \sum_{k=0}^{k_{max}} \left[ a^k \cos \left( 2\pi b^k \ast 0.5 \right) \right] + f_{bias_{11}}, \tag{A.19}
\]

\[a = 0.5, \quad b = 3, \quad k_{max} = 20, \quad z = (x - o) \ast M.\]

(20) Schwefel’s Problem 2.13 is as follows:
\[
f_{20}(x) = \sum_{j=1}^{D} (A_j - B_j \ast M) + f_{bias_{12}}, \tag{A.20}
\]

\[A_j = \sum_{j=1}^{D} \left( a_{ij} \sin a_j + b_{ij} \cos a_j \right), \]

\[B_j = \sum_{j=1}^{D} \left( a_{ij} \sin x_j + b_{ij} \cos x_j \right), \]

where $A$, $B$ are two $D \ast D$ 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. 71000172 and 71271110 and the National High Technology Research and Development Program of China (863 Program) (no. 2014AA052101-3).

References


