Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 927591, 17 pages
http://dx.doi.org/10.1155/2013/927591

Research Article

A Comparative Study of Improved Artificial Bee Colony Algorithms Applied to Multilevel Image Thresholding

Kanjana Charansiriphaisan, Sirapat Chiewchanwattana, and Khamron Sunat

Department of Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen 40002, Thailand

Correspondence should be addressed to Kanjana Charansiriphaisan; kanjanakku@gmail.com

Received 19 July 2013; Accepted 3 September 2013

Academic Editor: Erik Cuevas

Multilevel thresholding is a highly useful tool for the application of image segmentation. Otsu's method, a common exhaustive search for finding optimal thresholds, involves a high computational cost. There has been a lot of recent research into various metaheuristic searches in the area of optimization research. This paper analyses and discusses using a family of artificial bee colony algorithms, namely, the standard ABC, ABC/best/1, ABC/best/2, IABC/best/1, IABC/rand/1, and CABC, and some particle swarm optimization-based algorithms for searching multilevel thresholding. The strategy for an onlooker bee to select an employee bee was modified to serve our purposes. The metric measures, which are used to compare the algorithms, are the maximum number of function calls, successful rate, and successful performance. The ranking was performed by Friedman ranks. The experimental results showed that IABC/best/1 outperformed the other techniques when all of them were applied to multilevel image thresholding. Furthermore, the experiments confirmed that IABC/best/1 is a simple, general, and high performance algorithm.

1. Introduction

Image thresholding is a fundamental problem when trying to extract knowledge from an image. It has been effectively applied to many types of images. The homogeneous image regions are grouped together and separated from the rest. Thresholding can be bilevel or multilevel thresholding, and both of these types can be classified into parametric or nonparametric approaches. This problem area has been receiving attention from various research groups, and hundreds of algorithms have been proposed based on that research. There are five publications in various journals that have reviewed image thresholding [1–5] from which a comprehensive review can be accessed. The literature review revealed that Otsu's method [6] is a commonly used technique. It is a nonparametric approach and can solve both bilevel and multilevel thresholding. This method, which finds the optimal thresholds by maximizing the weighted sum of between-class variance of the classes, is called Otsu's function [6]. However, its computation is very time consuming; the complexity of the method grows exponentially with the number of thresholds.

The research carried out by Hammouche et al. has influenced the work carried out in this paper [7]. Their research focused on solving the image thresholding problem by merging Otsu's function with six metaheuristic techniques. The results of [7] concluded that the differential evolution (DE) [8] was the most efficient with respect to the quality of solution and because the particle swarm optimization (PSO) [9] converged more quickly. In this paper, a new family of metaheuristic techniques is used with Otsu's function to solve the image thresholding problem. The aim of this paper is to find a single metaheuristic technique that has the same characteristics of both DE and PSO. Some researchers proposed the same criterion as [7]. Kulkarni and Venayagamoorthy [10] formulated the sensor node localization to solve the image segmentation problem using PSO and the bacteria foraging algorithm (BF) [11]. Both of these algorithms are faster than Otsu's method for searching the optimal thresholds of multilevel image thresholding. In 2012, Ghamisi et al. [12] proposed methods for image segmentation based on two search techniques, namely, Darwinian particle swarm optimization (DPSO) [13] and the fractional-order Darwinian particle Swarm optimization (FODPSO) [14]. The results of FODPSO regarding the fitness value, standard deviation (STD), and CPU process time were better than those of its competitors, namely, DPSO, PSO, genetic algorithm (GA), and BF, especially when the segmentation level increased.

Artificial bee colony (ABC) is gaining popularity among the metaheuristic techniques because of its simplicity, generality, and remarkable performance [15–22]. It was also successfully applied to image thresholding [23, 24]. The evolution of ABC has seen continued improvements, and it has shown good performance levels in optimization problems. Balancing exploration and exploitation during a search is a successful criterion. In that regard, a family of the five modified search equations were chosen to generate the candidate solutions. The five equations were inspired by DE and GA. Those equations were embedded in the improved ABCs, that is, "ABC/best/1" and "ABC/rand/1" proposed in [25], "ABC/best/1" and "ABC/best/2" proposed in [26], and CABC proposed in [27]. Due to the fact that the equation used in ABC/best/1 in [25] is different from the equations used in ABC/best/1 in [26], this paper has renamed the algorithms in [25] to "IABC/best/1" and "IABC/rand/1," respectively. The improved ABCs showed outstanding performance in finding solutions to the benchmarking problems. However, their ability to search the optimal image thresholding is not known. To the best of the authors' knowledge, there is no paper published that reports this problem. The no free lunch theorem (NFLT) [28] is a major consideration in the area of optimization research. The theorem states that because the prior assumptions of the optimization problem are not known, no optimization algorithm can outperform any other algorithm. A general-purpose universal optimization algorithm has been proven to be theoretically impossible. One algorithm can outperform another if the specific problem is considered under reasonable restrictions. In this research, not only the five modified ABCs but also the standard ABC and other metaheuristic searches, that is, PSO, DPSO, and FODPSO, were implemented in the multilevel thresholding to compare their capabilities.

The remainder of the paper is organized as follows. In Section 2, the multilevel thresholding problem is formulated. Section 3 deals with the overview of ABC algorithm. Details of the various versions of the ABC algorithm are presented in Section 4. Performance evaluation and experimental results are presented in Section 5. The conclusions are given in Section 6.

2. Multilevel Thresholding Problem Formulation

Searching the optimal thresholding is a method to minimize or maximize the objective function of the separating classes. Otsu's method [6] is based on the maximization of the between-class variance. Given the image having size $H \times W$, where W is the width and H is the height, the pixels of a given picture are represented in L gray levels, and these levels are in the range $\{0, 1, 2, \ldots, L-1\}$. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_1 + n_2 + \cdots + n_L$. The gray-level histogram is normalized and regarded as a probability distribution:

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^{L} p_i = 1. \tag{1}$$

The total mean of the image can be defined as

$$\mu_T = \sum_{i=1}^{L} i \times p_i. \tag{2}$$

The multilevel thresholding with respect to the given $n-1$ threshold values t_j, $j = 1, \ldots, n-1$, can be performed as below:

$$F(x,y) = \begin{cases} 0, & f(x,y) \leq t_1, \\ \frac{1}{2}(t_1 + t_2), & t_1 < f(x,y) \leq t_2, \\ & \vdots \\ \frac{1}{2}(t_{n-2} + t_{n-1}), & t_{n-2} < f(x,y) \leq t_{n-1}, \\ L, & f(x,y) > t_{n-1}, \end{cases} \tag{3}$$

where (x, y) is the coordinate of a pixel, $f(x, y)$ denotes the intensity level of a pixel. The pixels of a given image will be divided into n classes $D_1, \ldots, D_n$ in this regard. If n is two, then $F(x, y)$ is called bilevel thresholding.

The optimal threshold is the one that maximizes the between-class variance, σ_B^2, which can be defined by Otsu's function:

$$\sigma_B^2 = \sum_{j=1}^{n} w_j (\mu_j - \mu_T)^2, \tag{4}$$

where j represents a specific class in such a way that w_j and μ_j are the probability of occurrence and the mean of class j, respectively. The probabilities of occurrence w_j of classes $D_1, \ldots, D_n$ are defined by

$$w_j = \begin{cases} \sum_{i=1}^{t_j} p_i, & j = 1, \\ \sum_{i=t_{j-1}+1}^{t_j} p_i, & 1 < j < n, \\ \sum_{i=t_{j-1}+1}^{L} p_i, & j = n. \end{cases} \tag{5}$$

The mean of each class μ_j can be given by

$$\mu_j = \begin{cases} \sum_{i=1}^{t_j} \frac{i \times p_i}{w_j}, & j = 1, \\ \sum_{i=t_{j-1}+1}^{t_j} \frac{i \times p_i}{w_j}, & 1 < j < n, \\ \sum_{i=t_{j-1}+1}^{L} \frac{i \times p_i}{w_j}, & j = n. \end{cases} \tag{6}$$

Thus, the n-level thresholding problem is transformed to an optimization problem. The process is to search for $n-1$ thresholds t_j that maximize the value φ, which is generally defined as

$$\varphi = \max_{1 < t_1 < \cdots < t_{n-1} < L} \sigma_B^2 (t_j). \tag{7}$$

The computational complexity of (7) can be as large as $O(L^{n-1})$ [7]. This is an NP-hard problem, which means that the exhaustive search will take a very long time if n is large.

3. Standard Artificial Bee Colony Algorithm

The artificial bee colony (ABC) algorithm was inspired by the foraging behaviors of bee colonies. This algorithm was first proposed by Karaboga [29], and it is referred to as the standard ABC. The ABC contains three groups of bees: employed bees, onlooker bees, and scout bees. The number of employed bees equals the number of onlooker bees. The employed bees search for the food around the food sources from their knowledge. Then, they pass their food information to the onlooker bees. Each onlooker bee tends to select good food sources founded by the employed bees. It then searches for more food around the selected food source. The bees that abandon their food sources will become scout bees, and they must now search for new food sources. A description of the algorithm is given below.

3.1. Initialization of the Population. The algorithm generates a group of food sources corresponding to the solutions in the search space. The SN-food sources, represented by $x = \{x_i \mid i = 1, 2, \ldots, SN\}$, are produced randomly within the range of the boundaries of the n-dimension variables:

$$x_{i,j} = x_{\min,j} + \operatorname{rand}(0,1)\left(x_{\max,j} - x_{\min,j}\right), \quad j = 1, 2, \ldots, n, \tag{8}$$

where $x_{\min,j}$ and $x_{\max,j}$ are the lower and upper bounds for the dimension j, respectively. The fitness of food sources x_i, $i = 1, 2, \ldots, SN$ will be evaluated. The counters that store the number of trials of each bee are set to 0 in this phase.

3.2. Employed Bees Phase. In the employed bees phase, each employed bee goes to the food source in its knowledge base and finds a neighboring food source. The neighboring food source v_i is defined by (9) as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j}\left(x_{i,j} - x_{k,j}\right), \tag{9}$$

where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, n\}$ are randomly chosen indexes: $k \neq j$ and $\phi_{i,j}$ is a uniform random number in the range $[-1, 1]$.

Fitness value for a minimization problem can be assigned to the solution v_i by (10) as follows:

$$\text{fitness}_i = \begin{cases} \dfrac{1}{1 + f_i}, & \text{if } f_i \geq 0, \\ 1 + \operatorname{abs}\left(f_i\right), & \text{if } f_i < 0, \end{cases} \tag{10}$$

where f_i is the cost value of the solution v_i. Once v_i is obtained, it will be evaluated and compared with x_i. If the fitness of v_i is equal to or better than that of x_i, x_i will be replaced by v_i, and v_i will become a new member of the population; otherwise, x_i is retained.

3.3. Calculating Probability Values Involved in Probabilistic Selection. After all of the employed bees complete their searches, the onlookers receive the information of the food sources from the employed bees. Then, they will each choose a food source to exploit depending on a probability related to the nectar amount of the food source (fitness values of the solution). That is to say, there may be more than one onlooker bee choosing the same food source if the source has a higher fitness. The probability is calculated according to (11) as follows:

$$p_i = \frac{\text{fitness}_i}{\sum_{j=1}^{SN} \text{fitness}_j}. \tag{11}$$

The lower the f_i, the more probability that the i food source is selected.

3.4. Onlooker Bees Phase. After food sources have been chosen, each onlooker bee x_i finds a new food source v_i in its neighbourhood, following (9), just like the employed bees do. If a new food source has better food than x_i, the new food source will replace x_i and become a new member in the population.

3.5. Scout Bees Phase. A greedy selection mechanism is employed between the old and candidate solutions of the employed and the onlooker bees phases. The trials counter of food will be reset to zero if the food source is improved; otherwise, its value will be incremented by one. In the scout bees phase, if the value of the trials counter of a food source is greater than a parameter, known as the "limit," the food source is abandoned and the bee becomes a scout bee. A new food source will be produced randomly in the search space using (8), as in the case of the initialization phase. The trials counter of the bee will be reset to zero.

4. The Various Artificial Bee Colony Algorithms

To enhance the performance of ABC, some modifications to the classical ABC algorithm have been made; the details of these modifications are as follows.

4.1. The Modified Probabilistic Selection Phase for the Maximization Problem. The original ABC was designed for minimization problems. However, multilevel thresholding is a maximization problem. Thus, some modifications are required. An onlooker bee will choose a food source in a different way to (10) and (11). That is, an onlooker bee will randomly choose a food source with a probability value p_i, which is calculated by the following form:

$$p_i = 0.9 \times \left(\frac{\sigma_{Bi}^2}{\sigma_{B\text{best}}^2}\right) + 0.1, \tag{12}$$

where σ_{Bi}^2 is obtained from Otsu's function of the solution i, $\sigma_{B\text{best}}^2$ is the best σ_B^2 among the employed bees population. As per (12), a higher between-class variance σ_B^2 computed by (4) obtains a higher probability. Thus, ABC can solve the maximization problem. That means that if the probability is calculated by (12), then the information from the elite employee bees that have higher between-class variances is passed to the onlooker bees.

4.2. Various Search Equations. Search equations are usually beneficial to the performance of metaheuristic techniques. The standard ABC is good at exploration but poor at exploitation [25–27]. The modified search equations were accordingly proposed to address this problem. A family of search equations have been suggested to create a new value of the randomly selected dimension $j \in \{1, 2, \ldots, n\}$ of the candidate food source v_i. The family of search equations are as follows.

Gao et al. [26] borrowed the mutation strategies of DE [8] to modify the solution search equation and proposed two global best-guided artificial bee colony algorithms; the search equations are shown as follows:

$$\text{ABC/best/1: } v_{i,j} = x_{\text{best},j} + \phi_{i,j}\left(x_{r1,j} - x_{r2,j}\right), \tag{13}$$

$$\begin{aligned}\text{ABC/best/2: } v_{i,j} = x_{\text{best},j} &+ \phi_{i,j}\left(x_{r1,j} - x_{r2,j}\right) \\ &+ \phi_{i,j}\left(x_{r3,j} - x_{r4,j}\right),\end{aligned} \tag{14}$$

where the indices $r1$, $r2$, $r3$, and $r4$ are mutually exclusive integers randomly chosen from $\{1, 2, \ldots, SN\}$ and different for the base index i, x_{best} is the best individual vector with the best fitness in the current population, and $\phi_{i,j}$ is a random number in the range $[-1, 1]$.

Gao and Liu [25] also included the other mutation strategies of DE [8] to ABC and proposed two strategies for improved artificial bee colony algorithms (IABC/best and IABC/rand), the equations for which are shown as follows:

$$\text{IABC/best/1: } v_{i,j} = x_{\text{best},j} + \phi_{i,j}\left(x_{i,j} - x_{r1,j}\right), \tag{15}$$

$$\text{IABC/rand/1: } v_{i,j} = x_{r1,j} + \phi_{i,j}\left(x_{i,j} - x_{r2,j}\right), \tag{16}$$

where the indices $r1$ and $r2$ are distinct integers uniformly chosen from the range $\{1, 2, \ldots, SN\}$, $r1 \neq i, r2 \neq i$, x_{best} is the best individual vector with the best fitness in the current population, and $\phi_{i,j}$ is a random number in the range $[-1, 1]$.

Gao et al. [27] brought the crossover operator of GA [30] to improve ABC. Their new search equation is as follows:

$$\text{CABC: } v_{i,j} = x_{r1,j} + \phi_{i,j}\left(x_{r1,j} - x_{r2,j}\right), \tag{17}$$

where the indices $r1$ and $r2$ are distinct integers uniformly chosen from the range $\{1, 2, \ldots, SN\}$, $r1 \neq i, r2 \neq i$, and $\phi_{i,j}$ is a random number in the range $[-1, 1]$. This search equation is similar to the crossover operator of GA and is called CABC.

4.3. Procedure of Various Artificial Bee Colony Algorithms for Multilevel Image Thresholding. This subsection shows the pseudocode of the five modified ABCs which apply to the image thresholding selection. The most important changes to these algorithms were the creation of a new candidate solution in each. Thus, the modified probabilistic selection phase and modified search equations from the previous subsections will be reformed in the pseudocode. In summary, IABC/best/1 is depicted as Algorithm 1. The four remaining algorithms have the same structure as of IABC/best/1, but line 09 and line 24 are different.

The Evaluate(·) functions at lines 02, 10, 25, and 41 are performed by Otsu's function, (4). Line 09 and line 24 of IABC/best/1 algorithm produce new candidates using the search equation (15). The algorithm can be performed as ABC/best/1, ABC/best/2, IABC/rand/1, or CABC algorithms by replacing (15) with (13), (14), (16), or (17), respectively. When the algorithm is applied to multilevel image thresholding, n is the number of thresholds. A food source represents a candidate n-threshold values. Once the algorithm terminates, the best food source is the solution, and the optimal n-threshold is the expected outcome. At this point in time, there are six algorithms where the standard ABC is included. Hence, the six-ABCs refers to them from now on.

5. Experiments and Results

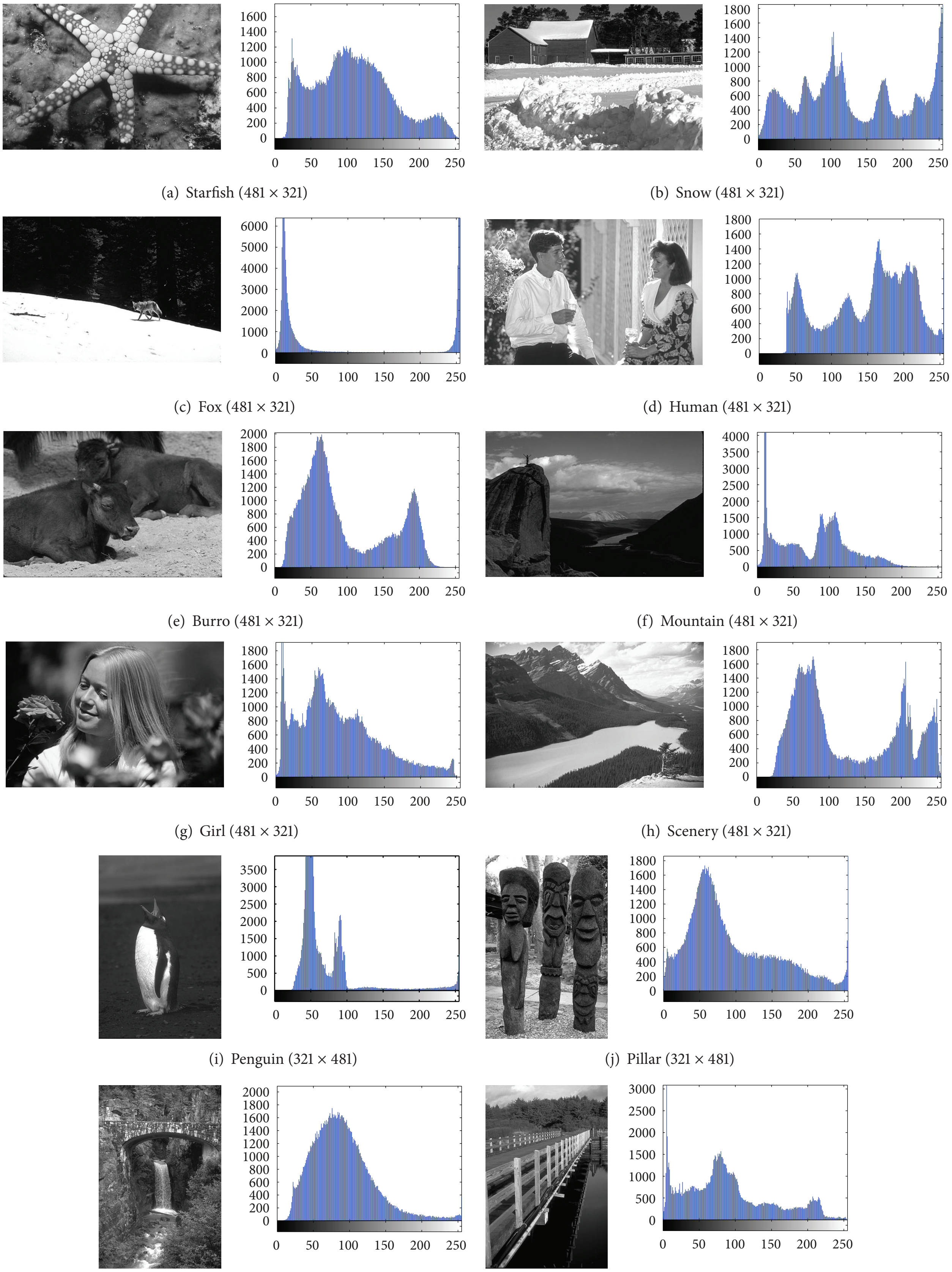
5.1. Experimental Setup. The multilevel thresholding problem deals with finding optimal thresholds within the range $[0, L-1]$ that maximize a fitness criterion. The dimension of the optimization problem is the number of thresholds, n, and the search space is $[0, L-1]^n$. Five new metaheuristic algorithms, that is, ABC/best/1, ABC/best/2, IABC/best/1, IABC/rand/1, and CABC, have never been applied to image thresholding before. These techniques are implemented and compared with the existing metaheuristics that perform image thresholding, that is, PSO, DPSO, FODPSO, and ABC algorithms. All methods were programmed in MATLAB on a personal computer with a 3.2 GHz CPU with 8 GB RAM running on a Microsoft Windows 7 system. Experiments were conducted on 12 real images. The images, namely starfish, snow, fox, human, burro, mountain, girl, scenery, penguin, pillar, waterfall, and bridge, were taken from the Berkeley Segmentation Dataset and Benchmark [31]. Each image has a unique gray level histogram. These original test images and their histograms are depicted in Figure 1. The segmented images, with regards to their optimal thresholds (listed in Table 2) are depicted in Figure 2. A subproblem refers to a process of an image thresholding with respect to a given number of thresholds. The numbers of thresholds investigated in the experiments were two to five. Thus, the total number of sub-problems is 12 × 4. Each sub-problem was repeated 50 times per image, and each time is called a run.

With reference to the six-ABCs and PSO algorithms, the objective function evaluation is computed for $N_p \times N_i$, where N_p is the population size and N_i is the number of generations. A PSO particle calls Otsu's function one time per generation. The population size in the PSO algorithm was set to 50. A bee in the six-ABCs calls Otsu's function two times per generation their numbers of food sources were set to a half of the PSO's size, 25. As for the DPSO and FODPSO algorithms comprising of 25 particles with 4 swarms, their objective function evaluations per generation are twice those of the six-ABCs and PSO. In this regard, the number of generations cannot be used as a metric for algorithm comparison, but the NFC is comparable. The other control parameters of PSO, DPSO, and FODPSO are tabulated in Table 1. The maximum number of generations was set to 50, 100, 150, and 150 when n is 2, 3, 4, and 5, respectively.

```
Initialize the population: $n$ is the number of the threshold. $SN$ is the population size.
    $x_{i,j} = x_{\min,j} + \text{rand}(0,1)(x_{\max,j} - x_{\min,j})$, $i = 1, 2, \ldots, SN$, $j = 1, 2, \ldots, n$.
Evaluate the population: $y = \text{Evaluate}(x)$ by Otsu's function, (4)
generation = 1, $\text{trial}_i = 0$, limit = 50, NFC = $SN$,
Repeat
/* Employed Bees Phase */
    For $i = 1$ to $SN$
        Randomly choose $x_{r1} \neq x_i$, $x_{r2} \neq x_i$, $x_{r1} \neq x_{r2}$ from current population
        Randomly choose $j$ from $\{1, 2, \ldots, n\}$
        $v_{i,j} = x_{\text{best},j} + \phi_{i,j}(x_{i,j} - x_{r1,j})$, (15)
        $y_{\text{new}} = \text{Evaluate}(v_i)$ by Otsu's function, (4), NFC = NFC + 1,
        If $y_{\text{new}} > y_i$
            $x_i = v_i$, $y_i = y_{\text{new}}$, $\text{trial}_i = 0$
        Else
            $\text{trial}_i = \text{trial}_i + 1$
        End If
    End For
    Calculate the probability values $p_i$ by (12)
/* Onlooker Bees Phase */
$i = 1$
For $t = 1$ to $SN$
    If $\text{rand}(0,1) < p_i$ then
        Randomly choose $x_{r1} \neq x_i$, $x_{r2} \neq x_i$, $x_{r1} \neq x_{r2}$ from current population
        Randomly choose $j$ from $\{1, 2, \ldots, n\}$
        $v_{i,j} = x_{\text{best},j} + \phi_{i,j}(x_{i,j} - x_{r1,j})$, (15)
        $y_{\text{new}} = \text{Evaluate}(v_i)$ by (4), NFC = NFC + 1,
        If $y_{\text{new}} > y_i$
            $x_i = v_i, y_i = y_{\text{new}}$, $\text{trial}_i = 0$
        Else
            $\text{trial}_i = \text{trial}_i + 1$
        End If
    End If
    $i = i + 1$
    If $i > SN$
        $i = 1$
    End If
End For
/* Scout Bees Phase */
For $i = 1$ to $SN$
    If $\text{trial}_i > \text{limit}$
        $x_{i,j} = x_{\min,j} + \text{rand}(0,1)(x_{\max,j} - x_{\min,j})$, $j = 1, 2, \ldots, n$, $\text{trial}_i = 0$
        $y_i = \text{Evaluate}(x_i)$ by (4), NFC = NFC + 1,
    End If
End For
Memorize the best solution achieved so far
generation = generation + 1
Until (generation = Maximum Generation)
```

Algorithm 1: (IABC/best/1 algorithm).

5.2. Comparing Strategies and Metrics. To minimize the effect of the stochastic nature of the algorithms on the metrics, the reported number of function calls for each sub-problem is the average or the maximum value over 50 runs. In this study, three metrics, namely, the maximum number of function calls ($\text{NFC}_{\max}$), successful rate (SR), and successful performance (SP), have been utilized to compare the algorithms. The convergence speed was compared by the measurement of the maximum number of function calls $\text{NFC}_{\max}$. A smaller $\text{NFC}_{\max}$ means higher convergence speed. The termination criterion of an algorithm is generally to find a value smaller than the value to reach (VTR) before reaching both the maximum number of function calls $\text{NFC}_{\max}$ and the maximum number of generations. The VTR in this research is the maximum between-class variance when the optimal thresholds were obtained from Otsu's method. Finding the $\text{NFC}_{\max}$ is a priority. Once all 50 runs were completed, the runs that reached the optimal solutions within the maximum



(a) Starfish (481 × 321)

(b) Snow (481 × 321)

(c) Fox (481 × 321)

(d) Human (481 × 321)

(e) Burro (481 × 321)

(f) Mountain (481 × 321)

(g) Girl (481 × 321)

(h) Scenery (481 × 321)

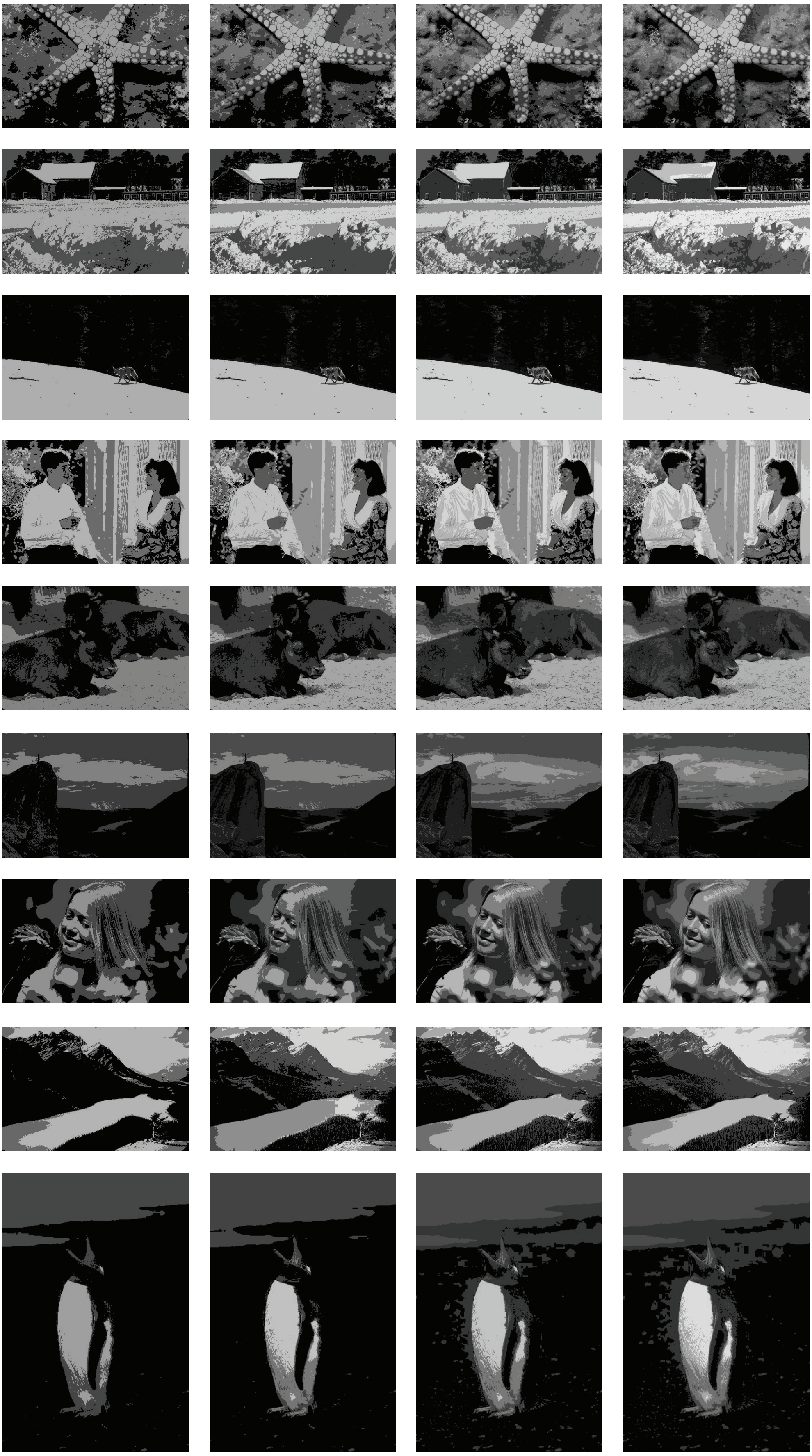
(i) Penguin (321 × 481)

(j) Pillar (321 × 481)

(k) Waterfall (321 × 481)

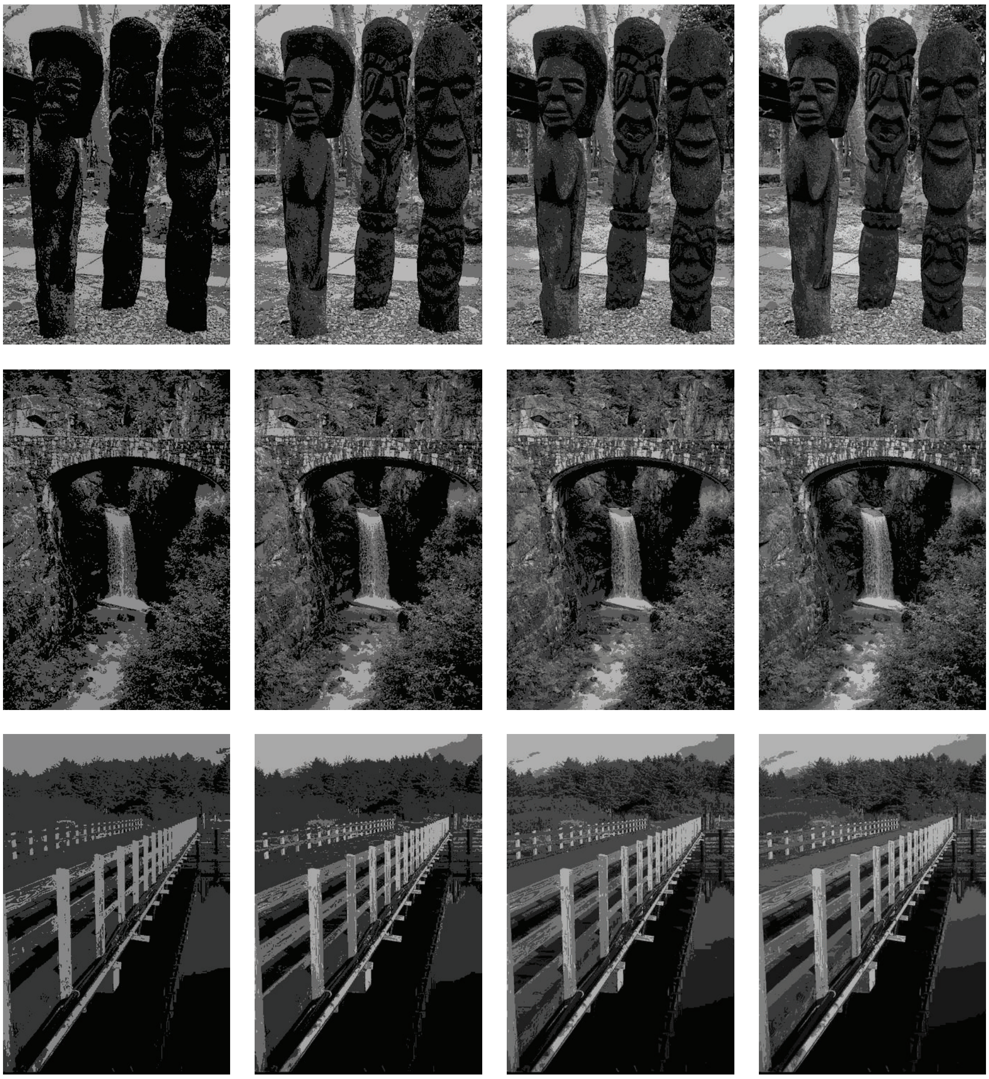
(l) Bridge (321 × 481)

FIGURE 1: The test images and corresponding histograms.



(a)

Figure 2: Continued.



(b)

Figure 2: The result of segmentation with 2, 3, 4, and 5 thresholds, respectively (from left to right).

number of generations were selected. These were called the successful runs. The maximum value of their NFCs represents the $\text{NFC}_{\max}$. The average of $\text{NFC}_{\max}$ represents the $\text{NFC}_{\overline{\max}}$.

The ratio for which the algorithm succeeds in reaching the VTR for each sub-problem is measured as the successful rate. SR is

$$\text{SR} = \frac{\text{number of successful runs}}{\text{total number of runs}}. \tag{18}$$

Furthermore, the average successful rate (SR_{avg}) over the entire data set per sub-problem is calculated as follows:

$$\text{SR}_{\text{avg}} = \frac{1}{\text{N}}\sum_{i=1}^{N}\text{SR}_i, \tag{19}$$

where SR_i represents the SR of each image of each sub-problem, and N represents the number of images. The highest SR and the lowest $\text{NFC}_{\max}$ are the ultimate goals in an optimization process. Therefore, two individual objectives must be considered simultaneously. In order to produce a single measure, called successful performance (SP), the two measures mentioned above have to be combined; this gives the following:

$$\text{SP} = \frac{\text{NFC}_{\max}}{\text{SR}}. \tag{20}$$

The SP is the main measure in judging which algorithm performs better than others.

5.2.1. Results of Otsu's Method. The VTR and optimal thresholds provided comparative experimental results as shown in columns 3 and 4 in Table 2. The results were as same as the optimal values of the nine metaheuristic techniques. However, the number of Otsu's function evaluations, in the case of an exhaustive search method based on Otsu's method, is $L!/((L-n)!n!)$, where L is the total number of gray-levels

Table 1: Essential parameters of the PSO, DPSO, and FODPSO.

Parameter	PSO	DPSO	FODPSO
Population	50	25	25
ρ_1	1.5	1.5	1.5
ρ_2	1.5	1.5	1.5
W	1.2	1.2	1.2
$V_{\max}$	2	2	2
$V_{\min}$	−2	−2	−2
$x_{\max}$	255	255	255
$x_{\min}$	0	0	0
Min population	—	10	10
Max population	—	50	50
Num of swarms	—	4	4
Min swarms	—	2	2
Max swarms	—	6	6
Stagnancy	—	10	10
Fractional coefficient	—	—	0.75

(usually 256) and n is the number of thresholds. For instance, when $L = 256$ and $n = 1$, the number of objective function evaluations is 256, and when $L = 256$ and $n = 2$, it is 32640. Thus, the computational complexity of Otsu's methods is $O(L^n)$ which grows exponentially with the number of thresholds. The computation time of a sub-problem with $n = 5$ took over 10 days to compute on our test machine.

5.2.2. Stability Analysis. In general, the metaheuristic methods are stochastic and random searching algorithms. The results are not absolutely identical in each run of the algorithm and are influenced by the searching ability of the algorithm. As a result, it is necessary to evaluate the stability of the algorithms. The comparison of the convergence gives us valuable information in terms of the ratio representing the successful rates (SR) in Table 2.

Table 2 presents the number of thresholds, the maximum between-class variance or VTR, the optimal threshold values, successful rates (SR) of the 50 independent runs, and the maximum number of function calls ($\mathrm{NFC}_{\max}$). The average successful rate for each sub-problem $\mathrm{SR}_{\mathrm{avg}}$ is shown in Table 3. The last row and respective column of Table 3 show the averages from all of the experiments. A higher SR means a higher stability. From the results, the stability of each algorithm can be sorted in decreasing order by $\mathrm{SR}_{\mathrm{avg}}$ into the following order: IABC/best/1, ABC/best/2, ABC/best/1, IABC/best/2, ABC, PSO, DPSO, CABC, and FODPSO. This means that the IABC/best/1 is the most stable metaheuristic algorithm when compared to the others.

5.2.3. Convergence Rate Comparison. The maximum numbers of NFC of the successful runs are listed in Table 2. A smaller $\mathrm{NFC}_{\max}$ means a higher convergence rate. The average of $\mathrm{NFC}_{\max}$ for each sub-problem is shown in Table 3. The last row and respective column of Table 3 show the average of each $\mathrm{NFC}_{\max}$ from all of the experiments. From the results, the convergence rate can be sorted in decreasing order by $\mathrm{NFC}_{\overline{\max}}$ into the following order: IABC/best/1, ABC/best/1, ABC/best/2, IABC/rand/1, CABC, PSO, ABC, DPSO, and FODPSO. This means that the IABC/best/1 has the highest convergent rate when compared to the other algorithms. The results also revealed that the $\mathrm{NFC}_{\max}$ increases significantly as the number of thresholds increases.

5.2.4. Performance Comparison. Scatter plots helped to reveal the relationships between the variables. There are four figures, one figure per threshold number; that is, Figure 3 is for $n = 2$, Figure 4 is for $n = 3$, Figure 5 is for $n = 4$, and Figure 6 is for $n = 5$. Each figure is constructed using two data sets, the successful rates (SR) versus the maximum number of function calls $\mathrm{NFC}_{\max}$ of all competitors from Table 2 and the average successful rates ($\mathrm{SR}_{\mathrm{avg}}$) versus the average of $\mathrm{NFC}_{\max}(\mathrm{NFC}_{\overline{\max}})$ from Table 3; the locations are pointed to using arrows. Those locations represent the average performances over twelve tested images of the algorithms. Some of the algorithm's labels are not shown because its $\mathrm{NFC}_{\max}$ is much larger than that of the others, and therefore, they are out of range. To interpret a figure, a lower $\mathrm{NFC}_{\max}$ and a larger SR imply a higher performing algorithm in the search for optimal thresholds. This relationship can be seen as a point at the topleft of each scatter graph. Due to the fact that all of the scatter graph results are so similar, only information regarding the IABC/best/1 will be interpreted. In Figures 3 to 6, it was realized that the positions of IABC/best/1 are closer to the left-top corners than those of the other techniques for all the figures. This means that IABC/best/1 outperformed its competitors.

However, if there is a lot of data, the conclusion from the previous paragraph may not be so clear. Therefore, a new tool is required. The successful performance (SP) is a measure that considers the maximum number of function calls and the successful rate simultaneously, as listed in Table 4. The best SPs are highlighted in boldface. To rank the algorithm by successful performance, an average ranking of Friedman ranks [32] was used to compare the performance. The average ranks are shown in the last row of Table 4. A smaller rank means a better performance. We noticed that the SPs of the nine algorithms can be sorted in decreasing order by average ranking into the following order: IABC/best/1, ABC/best/1, ABC/best/2, IABC/rand/1, CABC, PSO, ABC, DPSO, and FODPSO. The best average ranking was obtained by the IABC/best/1 algorithm, which outperformed the other eight algorithms.

6. Conclusions

This paper adopted the ABC algorithm to solve the maximization problem by modifying the strategy of the elite employee bees selection. Five methods for multilevel segmentation of images were presented, namely, ABC/best/1, ABC/best/2, IABC/best/1, IABC/rand/1, and CABC. These methods were used to solve the Otsu problem for examining multilevel threshold values of images and to overcome the disadvantages of previous metaheuristic methods in terms of

Table 2: The successful rate (SR) and maximum number of function calls (NFC_{max}) of the nine methods.

Image	n	Between-class variance (a.k.a. VTR)	Thresholds	PSO		DPSO		FODPSO		ABC		ABC/best/1		ABC/best/2		IABC/best/1		IABC/rand/1		CABC	
				SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}
Starfish	2	2546.885	85, 157	1.00	1350	1.00	3200	1.00	4200	1.00	1150	0.96	500	1.00	650	1.00	450	1.00	1000	0.94	800
	3	2779.925	68, 119, 177	1.00	2050	1.00	5600	0.92	8600	1.00	1900	1.00	850	1.00	950	1.00	800	1.00	1250	1.00	1000
	4	2865.707	60, 101, 138, 187	1.00	4550	0.98	11100	0.60	14700	1.00	6500	1.00	2000	1.00	3350	1.00	2200	1.00	3900	0.96	5350
	5	2912.859	52, 86, 117, 150, 194	0.72	7450	0.56	15000	0.26	12900	0.66	7250	0.88	5150	0.90	7500	1.00	4600	0.76	7300	0.76	6650
Snow	2	5261.705	80, 169	1.00	1150	1.00	3500	1.00	6700	1.00	1500	1.00	700	1.00	950	1.00	550	1.00	1100	0.98	1000
	3	5624.289	71, 139, 207	1.00	2150	1.00	8800	0.94	6600	1.00	2150	1.00	750	1.00	900	1.00	750	1.00	1150	0.94	1150
	4	5729.116	50, 92, 144, 208	0.98	4150	0.98	12000	0.68	15000	1.00	3500	1.00	1550	1.00	2150	1.00	2250	1.00	4050	1.00	4250
	5	5785.138	49, 91, 140, 192, 231	0.58	7100	0.60	14900	0.12	14900	0.98	7200	1.00	2700	1.00	3900	1.00	2550	1.00	6550	0.92	7000
Fox	2	12937.730	54, 173	1.00	550	1.00	2200	1.00	3100	1.00	650	1.00	300	1.00	400	1.00	350	1.00	500	1.00	450
	3	12973.370	28, 85, 190	1.00	1200	1.00	4500	0.98	7100	1.00	2050	1.00	700	1.00	1100	1.00	850	1.00	1750	0.76	1400
	4	12990.700	23, 58, 123, 208	0.96	2500	0.86	8200	0.60	8300	1.00	3150	0.88	1250	1.00	1950	1.00	1350	1.00	2400	0.82	2450
	5	12998.990	19, 40, 79, 139, 214	0.98	3550	0.02	12300	0.88	8100	0.98	5500	1.00	2450	1.00	3300	1.00	2200	0.98	6400	0.80	5650
Human	2	3064.796	108, 179	1.00	1350	1.00	3300	1.00	4200	1.00	1050	1.00	350	1.00	650	1.00	550	1.00	750	0.96	950
	3	3270.671	89, 145, 194	1.00	2650	1.00	4700	0.94	9500	1.00	2250	1.00	750	1.00	1000	1.00	800	1.00	1100	0.92	950
	4	3336.302	87, 140, 180, 213	0.98	4200	0.90	11200	0.40	15000	0.98	6200	0.98	7500	1.00	4000	1.00	3800	0.74	7000	0.56	7450
	5	3373.672	74, 110, 147, 183, 215	0.70	7050	0.48	14300	0.24	14600	0.66	7350	0.84	3600	1.00	5100	1.00	4600	0.92	4450	0.40	6800
Burro	2	3161.277	64, 131	1.00	1400	1.00	3100	0.98	5800	1.00	1750	0.96	450	1.00	1100	1.00	850	1.00	1400	0.98	1800
	3	3293.718	57, 107, 165	1.00	3150	1.00	6400	0.98	8800	1.00	2100	1.00	850	1.00	1200	1.00	1050	1.00	1400	1.00	1150
	4	3353.639	49, 79, 122, 171	0.96	3800	0.88	13000	0.34	15000	1.00	3850	0.72	1400	0.96	2200	1.00	1700	1.00	2650	0.98	2900
	5	3382.261	45, 70, 101, 140, 177	0.66	7200	0.78	14700	0.66	12500	0.44	7500	0.90	7050	0.82	6250	0.92	7000	0.48	7450	0.20	6950
Mountain	2	2372.923	61, 128	1.00	1750	1.00	2800	1.00	5400	1.00	1100	0.96	350	1.00	650	1.00	450	1.00	600	0.94	550
	3	2496.113	33, 77, 131	1.00	2150	1.00	6300	0.96	9300	1.00	2400	1.00	1000	1.00	1250	1.00	800	1.00	1950	1.00	1800
	4	2551.955	33, 73, 109, 147	0.98	5750	0.96	13800	0.36	13700	1.00	4600	0.96	3000	1.00	2650	1.00	2000	1.00	2800	0.94	4500
	5	2580.336	32, 69, 99, 125, 159	0.70	7200	0.66	14800	0.28	14700	1.00	6500	0.98	1950	1.00	3200	1.00	2050	1.00	4300	0.72	6450
Girl	2	2639.526	72, 145	1.00	1350	1.00	3200	1.00	4900	1.00	1400	1.00	500	1.00	700	1.00	500	1.00	900	0.98	750
	3	2867.882	46, 96, 161	1.00	1900	1.00	6300	0.88	10000	1.00	2250	1.00	800	1.00	1200	1.00	800	1.00	1650	0.76	1800
	4	2971.761	42, 84, 128, 182	0.96	6950	0.94	14400	0.52	14900	1.00	4050	1.00	2500	1.00	2350	1.00	1900	1.00	2600	0.94	4650
	5	3022.602	37, 70, 102, 140, 189	0.74	7450	0.78	14500	0.66	14800	0.82	7350	0.98	2850	1.00	6900	1.00	4700	0.84	7450	0.48	6500
Scenery	2	4686.124	105, 180	1.00	1150	1.00	4100	1.00	6900	1.00	2150	1.00	750	1.00	1050	1.00	800	1.00	1550	0.96	3800
	3	4837.868	76, 142, 211	1.00	3500	1.00	8200	0.84	9700	0.98	4000	0.98	1350	1.00	3150	1.00	1450	0.96	4950	0.92	4050
	4	4944.489	67, 111, 169, 218	0.98	5050	0.96	11200	0.48	12400	1.00	3050	0.98	1800	1.00	1800	1.00	1300	1.00	2100	0.98	2200
	5	4983.329	64, 98, 141, 184, 221	0.84	7000	0.58	14400	0.22	13800	0.80	7500	0.98	7100	0.96	7300	0.98	4750	0.56	7200	0.58	7300
Penguin	2	1823.215	71, 159	1.00	1200	1.00	3100	1.00	3400	1.00	800	1.00	350	1.00	450	1.00	350	1.00	550	1.00	500
	3	1892.583	68, 119, 194	1.00	2200	1.00	4700	0.92	5600	1.00	1850	0.98	700	1.00	900	1.00	800	1.00	1100	0.98	1050
	4	1915.004	54, 76, 121, 195	0.54	4300	0.62	15000	0.22	13800	0.56	6750	0.70	5000	0.84	5550	0.88	6350	0.80	7300	0.76	7400
	5	1932.441	53, 75, 114, 167, 220	0.82	7250	0.56	12500	0.20	13200	0.90	6300	0.96	2200	1.00	3200	1.00	2150	1.00	3250	0.94	3600
Pillar	2	3306.612	87, 164	1.00	1050	1.00	3700	1.00	4700	1.00	1400	0.98	600	1.00	850	1.00	550	1.00	1200	0.94	1450
	3	3534.700	58, 113, 184	1.00	2450	1.00	5900	0.74	9000	1.00	4350	1.00	1600	1.00	2000	1.00	1300	1.00	2050	0.98	2000
	4	3651.357	48, 90, 141, 200	0.94	4900	0.96	10500	0.66	14000	0.84	6250	0.98	4950	0.98	3500	1.00	4050	0.98	4300	0.82	4300
	5	3714.997	40, 73, 111, 156, 208	0.80	6900	0.80	14300	0.54	11600	0.62	7500	0.90	5850	0.96	6350	0.98	5650	0.90	7150	0.84	6200

Table 2: Continued.

Image	n	Between-class variance (a.k.a. VTR)	Thresholds	PSO		DPSO		FODPSO		ABC		ABC/best/1		ABC/best/2		IABC/best/1		IABC/rand/1		CABC	
				SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}	SR	NFC_{max}
Waterfall	2	1444.433	81, 143	1.00	1050	1.00	3100	1.00	4000	1.00	1000	1.00	400	1.00	600	1.00	500	1.00	700	1.00	700
	3	1597.070	68, 108, 167	1.00	4850	1.00	7600	0.84	9800	1.00	2650	1.00	1000	1.00	1300	1.00	1200	1.00	1950	0.92	1750
	4	1669.214	59, 91, 126, 181	0.92	6750	0.90	13900	0.56	14200	1.00	5350	0.90	4100	1.00	2700	1.00	2350	1.00	3750	0.94	6600
	5	1708.521	53, 80, 106, 138, 190	0.60	6800	0.64	14000	0.58	13100	0.46	7150	0.96	3300	0.94	5900	1.00	4450	0.76	6550	0.56	6600
Bridge	2	3383.821	54, 135	1.00	1400	1.00	2500	1.00	5000	1.00	900	0.98	450	1.00	450	1.00	450	1.00	700	0.98	700
	3	3567.597	49, 108, 169	1.00	3650	0.98	6300	0.70	9300	1.00	2150	0.98	900	1.00	1550	1.00	900	1.00	1800	1.00	1650
	4	3660.335	34, 73, 117, 174	0.96	5450	0.92	13100	0.64	14000	0.80	7000	0.92	4800	1.00	4200	1.00	3400	0.90	4750	0.70	3700
	5	3710.264	26, 59, 90, 128, 179	0.70	6000	0.66	14900	0.68	14100	0.94	6950	0.94	3550	0.96	5000	0.96	3750	0.78	7350	0.22	6950

TABLE 3: The average of successful rates (SR_{avg}) and average of maximum number of function calls ($\text{NFC}_{\overline{\text{max}}}$) of the nine methods.

The number of	PSO		DPSO		FODPSO		ABC		ABC/best/1		ABC/best/2		IABC/best/1		IABC/rand/1		CABC	
thresholds (n)	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$	SR_{avg}	$\text{NFC}_{\overline{\text{max}}}$
2	1	1229.16	1	3150	0.99	4858.33	1	1237.50	0.98	475.00	1	708.33	1	529.16	1	912.50	0.97	1120.83
3	1	2658.33	0.99	6275	0.88	8608.33	0.99	2508.33	0.99	937.50	1	1375.00	1	958.33	0.99	1841.66	0.93	1645.83
4	0.93	4862.5	0.90	12283.33	0.50	13750.00	0.93	5020.83	0.91	3320.83	0.98	3033.33	0.99	2720.83	0.95	3966.66	0.86	4645.83
5	0.73	6745.83	0.59	14216.67	0.44	13191.67	0.77	7004.16	0.94	3979.16	0.96	5325.00	0.98	4037.50	0.83	6283.33	0.61	6387.50
Averages of the entire experiments	0.91	3873.95	0.87	8981.25	0.70	10102.08	0.92	3942.70	0.96	2178.12	0.98	2610.41	**0.99**	**2061.45**	0.94	3251.04	0.84	3450.00

Table 4: The successful performance (SP) and ranks of the nine methods.

Image	*n*	PSO		DPSO		FODPSO		ABC		ABC/best/1		ABC/best/2		IABC/best/1		IABC/rand/1		CABC	
		SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank
Starfish	2	1350	7	3200	8	4200	9	1150	6	520.83	2	650	3	**450**	**1**	1000	5	851.06	4
	3	2050	7	5600	8	9347.83	9	1900	6	850	2	950	3	**800**	**1**	1250	5	1000	4
	4	4550	5	11326.53	8	24500	9	6500	7	**2000**	**1**	3350	3	2200	2	3900	4	5572.92	6
	5	10347.22	6	26785.71	8	49615.38	9	10984.85	7	5852.27	2	8333.33	3	**4600**	**1**	9605.26	5	8750	4
Snow	2	1150	6	3500	8	6700	9	1500	7	700	2	950	3	**550**	**1**	1100	5	1020.41	4
	3	2150	6.5	8800	9	7021.28	8	2150	6.5	**750**	**1.5**	900	3	**750**	**1.5**	1150	4	1223.40	5
	4	4234.69	6	12244.90	8	22058.82	9	3500	4	**1550**	**1**	2150	2	2250	3	4050	5	4250	7
	5	12241.38	7	24833.33	8	124166.70	9	7346.94	5	2700	2	3900	3	**2550**	**1**	6550	4	7608.70	6
Fox	2	550	6	2200	8	3100	9	650	7	**300**	**1**	400	3	350	2	500	5	450	4
	3	1200	4	4500	8	7244.89	9	2050	7	**700**	**1**	1100	3	850	2	1750	5	1842.11	6
	4	2604.17	5	9534.88	8	13833.33	9	3150	7	1420.46	2	1950	3	**1350**	**1**	2400	4	2987.81	6
	5	3622.45	4	615000	8	9204.55	9	5612.25	5	2450	2	3300	3	**2200**	**1**	6530.61	6	7062.50	7
Human	2	1350	7	3300	8	4200	9	1050	6	**350**	**1**	650	3	550	2	750	4	989.58	5
	3	2650	7	4700	8	10106.38	9	2250	6	**750**	**1**	1000	3	800	2	1100	5	1032.61	4
	4	4285.71	3	12444.44	7	37500	9	6326.53	4	7653.06	5	4000	2	**3800**	**1**	9459.46	6	13303.57	8
	5	10071.43	5	29791.67	8	60833.33	9	11136.36	6	**4285.71**	**1**	5100	4	4600	2	4836.96	3	17000	7
Burro	2	1400	4.5	3100	8	5918.37	9	1750	6	**468.75**	**1**	1100	3	850	2	1400	4.5	1836.74	7
	3	3150	7	6400	8	8979.59	9	2100	6	**850**	**1**	1200	4	1050	2	1400	5	1150	3
	4	3958.33	7	14772.73	8	44117.65	9	3850	6	1944.44	2	2291.67	3	**1700**	**1**	2650	4	2959.18	5
	5	10909.09	4	18846.15	7	18939.39	8	17045.45	6	7833.33	3	7621.95	2	**7608.70**	**1**	15520.83	5	34750	9
Mountain	2	1750	7	2800	8	5400	9	1100	6	**364.58**	**1**	650	5	450	2	600	4	585.11	3
	3	2150	6	6300	8	9687.50	9	2400	7	1000	2	1250	3	**800**	**1**	1950	5	1800	4
	4	5867.35	7	14375	8	38055.56	9	4600	5	3125	4	2650	2	**2000**	**1**	2800	3	4787.23	6
	5	10285.71	7	22424.24	8	52500	9	6500	5	**1989.80**	**1**	3200	3	2050	2	4300	4	8958.33	6
Girl	2	1350	6	3200	8	4900	9	1400	7	**500**	**1.5**	700	3	**500**	**1.5**	900	5	765.31	4
	3	1900	5	6300	8	11363.64	9	2250	6	**800**	**1.5**	1200	3	**800**	**1.5**	1650	4	2368.42	7
	4	7239.58	7	15319.15	8	28653.85	9	4050	5	2500	3	2350	2	**1900**	**1**	2600	4	4946.81	6
	5	10067.57	6	18589.74	8	22424.24	9	8963.42	5	**2908.16**	**1**	6900	3	4700	2	8869.05	4	13541.67	7
Scenery	2	1150	4	4100	8	6900	9	2150	6	**750**	**1**	1050	3	800	2	1550	5	3958.33	7
	3	3500	4	8200	8	11547.62	9	4081.63	5	**1377.55**	**1**	3150	3	1450	2	5156.25	7	4402.17	6
	4	5153.06	7	11666.67	8	25833.33	9	3050	6	1836.74	3	1800	2	**1300**	**1**	2100	4	2244.90	5
	5	8333.33	4	24827.59	8	62727.27	9	9375	5	7244.90	2	7604.17	3	**4846.94**	**1**	12857.14	7	12586.21	6
Penguin	2	1200	7	3100	8	3400	9	800	6	**350**	**1.5**	450	3	**350**	**1.5**	550	5	500	4
	3	2200	7	4700	8	6086.96	9	1850	6	**714.29**	**1**	900	3	800	2	1100	5	1071.43	4
	4	7962.96	4	24193.55	8	62727.27	9	12053.57	7	7142.86	2	**6607.14**	**1**	7215.91	3	9125	5	9736.84	6
	5	8841.46	7	22321.43	8	66000	9	7000	6	2291.67	2	3200	3	**2150**	**1**	3250	4	3829.79	5

Table 4: Continued.

Image	n	PSO		DPSO		FODPSO		ABC		ABC/best/1		ABC/best/2		IABC/best/1		IABC/rand/1		CABC	
		SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank	SP	Rank
Pillar	2	1050	4	3700	8	4700	9	1400	6	612.25	2	850	3	**550**	**1**	1200	5	1542.55	7
	3	2450	6	5900	8	12162.16	9	4350	7	1600	2	2000	3	**1300**	**1**	2050	5	2040.82	4
	4	5212.77	5	10937.50	8	21212.12	9	7440.48	7	5051.02	4	**3571.43**	**1**	4050	2	4387.76	3	5243.90	6
	5	8625	6	17875	8	21481.48	9	12096.77	7	6500	2	6614.58	3	**5765.31**	**1**	7944.44	5	7380.95	4
Waterfall	2	1050	7	3100	8	4000	9	1000	6	**400**	**1**	600	3	500	2	700	4.5	700	4.5
	3	4850	7	7600	8	11666.67	9	2650	6	**1000**	**1**	1300	3	1200	2	1950	5	1902.17	4
	4	7336.96	7	15444.44	8	25357.14	9	5350	5	4555.56	4	2700	2	**2350**	**1**	3750	3	7021.28	6
	5	11333.33	5	21875	8	22586.21	9	15543.48	7	**3437.50**	**1**	6276.60	3	4450	2	8618.42	4	11785.71	6
Bridge	2	1400	7	2500	8	5000	9	900	6	459.18	3	**450**	**1.5**	**450**	**1.5**	700	4	714.29	5
	3	3650	7	6428.57	8	13285.71	9	2150	6	918.37	2	1550	3	**900**	**1**	1800	5	1650	4
	4	5677.08	6	14239.13	8	21875	9	8750	7	5217.39	3	4200	2	**3400**	**1**	5277.78	4	5285.71	5
	5	8571.43	5	22575.76	8	20735.29	7	7393.62	4	**3776.60**	**1**	5208.33	3	3906.25	2	9423.08	6	31590.91	9
Average rank (lower is better)			5.85		7.98		8.9167		5.99		1.85		2.802		**1.53**		4.625		5.45

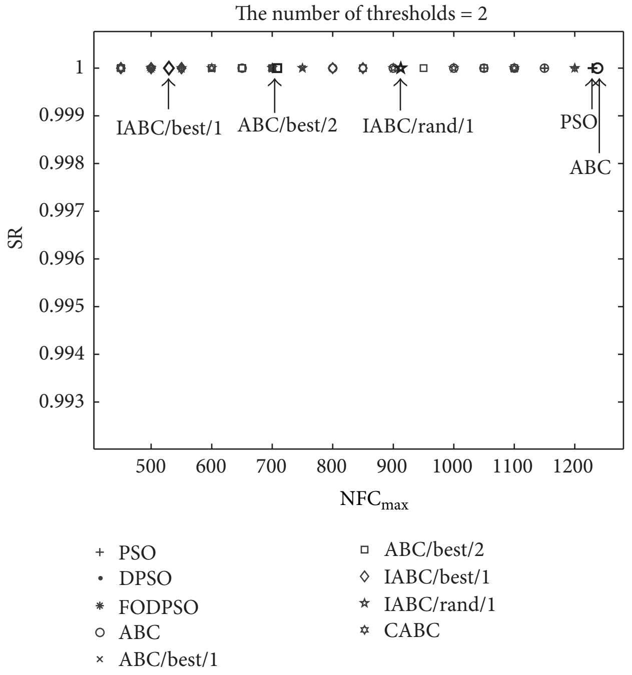


Figure 3: The successful rate (SR) and maximum number of function calls (NFC$_{max}$) of the nine methods with the number of thresholds = 2.

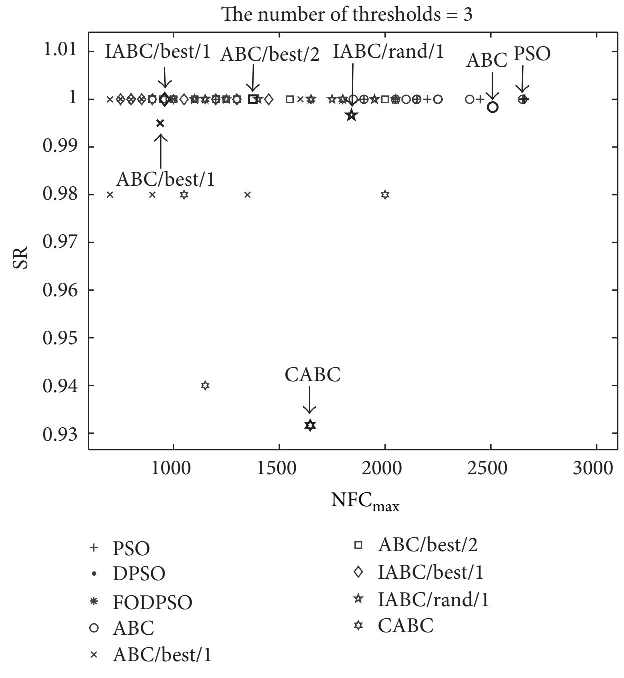


Figure 4: The successful rate (SR) and maximum number of function calls (NFC$_{max}$) of the nine methods with the number of thresholds = 3.

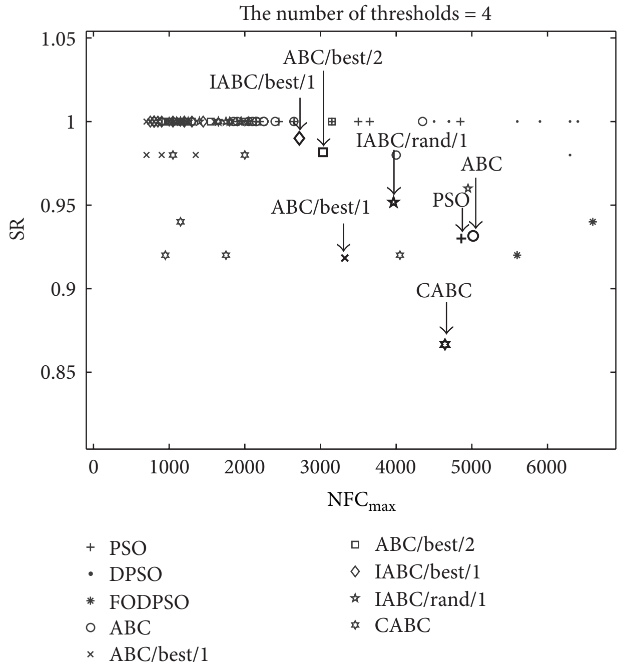


Figure 5: The successful rate (SR) and maximum number of function calls (NFC$_{max}$) of the nine methods with the number of thresholds = 4.

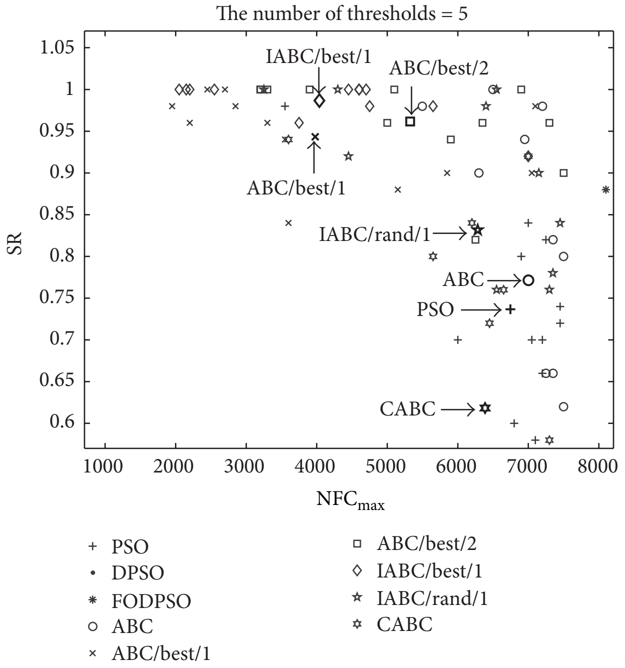


Figure 6: The successful rate (SR) and maximum number of function calls (NFC$_{max}$) of the nine different methods with the number of thresholds = 5.

exploration and exploitation. The performance of the algorithms was tested with 12 standard test images and compared with the ABC, PSO, DPSO, and FOPSO methods. The ranking was performed by Friedman ranks. The segmentation results obtained from various test images showed that the IABC/best/1 algorithm widely outperformed the other algorithms in terms of successful performance, computation efficiency, and stability. Another advantage of the ABC algorithm is that it has fewer control parameters than those of PSO. The experiments confirmed that IABC/best/1 is a simple, general, and high performance algorithm. Future research is to be carried out to test the feasibility of IABC/best/1 algorithm for various types of image processing applications such as remote sensing applications, automatic target recognition, and complex document analysis.

Conflict of Interests

The authors do not have any conflict of interests with the content of the paper.

Acknowledgments

This work was supported by the Higher Education Research Promotion and National Research University Project of Thailand, Office of the Higher Education Commission, through the Cluster of Research to Enhance the Quality of Basic Education.

References

[1] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *IEEE Transactions on Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.

[2] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[3] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.

[4] C.-I. Chang, Y. Du, J. Wang, S.-M. Guo, and P. D. Thouin, "Survey and comparative analysis of entropy and relative entropy thresholding techniques," *IEE Proceedings-vision Image and Signal Processing*, vol. 153, no. 6, pp. 837–850, 2006.

[5] P. Lenkiewicz, M. Pereira, M. M. Freire, and J. Fernandes, "Techniques for medical image segmentation: review of the most popular approaches," *Biomedical Diagnostics and Clinical Technologies*, 2011.

[6] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[7] K. Hammouche, M. Diaf, and P. Siarry, "A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 5, pp. 676–688, 2010.

[8] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," Tech. Rep. TR-95-012, ICSI, 1995.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks (ICNN '95)*, vol. 4, pp. 1942–1948, Perth, Australia, December 1995.

[10] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 40, no. 6, pp. 663–675, 2010.

[11] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.

[12] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, "An efficient method for segmentation of image based on fractional calculus and natural selection," *Expert Systems with Applications*, vol. 39, pp. 12407–12417, 2012.

[13] J. Tillett, T. M. Rao, F. Sahin, R. Rao, and S. Brockport, "Darwinian particle swarm optimization," in *Proceedings of the 2nd Indian International Conference on Artificial Intelligence*, pp. 1474–1487, 2005.

[14] M. S. Couceiro, R. P. Rocha, N. M. F. Ferreira, and J. A. T. Machado, "Introducing the fractional-order Darwinian PSO," *Signal Image and Video Processing*, vol. 6, no. 3, pp. 343–350, 2012.

[15] 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.

[16] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[17] N. Karaboga, "A new design method based on Artificial Bee Colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.

[18] A. Singh, "An Artificial Bee Colony algorithm for the leaf-constrained minimum spanning tree problem," *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 625–631, 2009.

[19] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 652–657, 2011.

[20] C. Zhang, D. Ouyang, and J. Ning, "An Artificial Bee Colony approach for clustering," *Expert Systems with Applications*, vol. 37, no. 7, pp. 4761–4767, 2010.

[21] Q.-K. Pan, M. F. Tasgetiren, P. N. Suganthan, and T. J. Chua, "A discrete Artificial Bee Colony algorithm for the lot-streaming flow shop scheduling problem," *Information Sciences*, vol. 181, no. 12, pp. 2455–2468, 2011.

[22] H. Shayeghi and A. Ghasemi, "Market based LFC design using Artificial Bee Colony," *International Journal on Technical and Physical Problems of Engineering*, vol. 3, no. 6, pp. 1–10, 2011.

[23] M.-H. Horng, "Multilevel thresholding selection based on the Artificial Bee Colony algorithm for image segmentation," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13785–13791, 2011.

[24] E. Cuevas, F. Sención, D. Zaldivar, M. Pérez-Cisneros, and H. Sossa, "A multi-threshold segmentation approach based on Artificial Bee Colony optimization," *Applied Intelligence*, vol. 37, no. 3, pp. 321–336, 2011.

[25] W. Gao and S. Liu, "Improved Artificial Bee Colony algorithm for global optimization," *Information Processing Letters*, vol. 111, no. 17, pp. 871–882, 2011.

[26] W. Gao, S. Liu, and L. Huang, "A global best Artificial Bee Colony algorithm for global optimization," *Journal of Computational and Applied Mathematics*, vol. 236, no. 11, pp. 2741–2753, 2012.

[27] W. Gao, S. Liu, and L. Huang, "A novel Artificial Bee Colony algorithm based on modified search equation and orthogonal learning," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 99, pp. 1–14, 2013.

[28] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[29] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.

[30] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[31] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 2, pp. 416–423, Vancouver, Canada, July 2001.

[32] T. M. Beasley and B. D. Zumbo, "Comparison of aligned Friedman rank and parametric methods for testing interactions in split-plot designs," *Computational Statistics & Data Analysis*, vol. 42, no. 4, pp. 569–593, 2003.