Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 1578056, 12 pages
http://dx.doi.org/10.1155/2016/1578056

Research Article

Multilevel Image Segmentation Based on an Improved Firefly Algorithm

Kai Chen, Yifan Zhou, Zhisheng Zhang, Min Dai, Yuan Chao, and Jinfei Shi

Mechanical Engineering School, Southeast University, Nanjing 211189, China

Correspondence should be addressed to Zhisheng Zhang; oldbc@seu.edu.cn

Received 1 December 2015; Revised 17 January 2016; Accepted 24 January 2016

Academic Editor: Pasquale Memmolo

Multilevel image segmentation is time-consuming and involves large computation. The firefly algorithm has been applied to enhancing the efficiency of multilevel image segmentation. However, in some cases, firefly algorithm is easily trapped into local optima. In this paper, an improved firefly algorithm (IFA) is proposed to search multilevel thresholds. In IFA, in order to help fireflies escape from local optima and accelerate the convergence, two strategies (i.e., diversity enhancing strategy with Cauchy mutation and neighborhood strategy) are proposed and adaptively chosen according to different stagnation stations. The proposed IFA is compared with three benchmark optimal algorithms, that is, Darwinian particle swarm optimization, hybrid differential evolution optimization, and firefly algorithm. The experimental results show that the proposed method can efficiently segment multilevel images and obtain better performance than the other three methods.

1. Introduction

Image segmentation is an indispensable part of computer vision, which directly influences the quality of feature extraction and classification. In the past few decades, many image segmentation methods have been proposed and applied in different fields [1–4]. Thresholding is one of the widely used techniques in image segmentation. The principle of thresholding is finding an adequate threshold to segment object from background through certain criteria, such as Otsu thresholding [5], maximum entropy thresholding [6], minimum error thresholding [7], and 2D maximum entropy thresholding [8]. In the case of segmenting several objects from background, thresholding technique needs to be extended to multilevel thresholding. However, large amount of calculation and long computation time occur when exhaustively searching multilevel thresholds.

To overcome the exhaustive problems, several metaheuristics optimal algorithms were applied to multilevel thresholding. Hammouche et al. [9] presented optimal thresholding using genetic algorithms. Ghamisi et al. [10] developed particle swarm optimization (PSO), Darwinian particle swarm optimization (DPSO), and fractional-order Darwinian particle swarm optimization for determining thresholds. Liang et al. [11] designed an ant colony optimization segmentation algorithm for solving multilevel Otsu problem. Sathya and Kayalvizhi [12] applied bacterial foraging into finding thresholds for maximizing Kapur's and Otsu's objective functions. Cuevas et al. [13] used differential evolution optimization (DE) to find multilevel thresholds. Osuna-Enciso et al. [14] presented a comparison of three optimization algorithms for selecting multilevel thresholds. Nevertheless, with the number of thresholds increasing, these metaheuristics optimal algorithms cannot find the balance of global search and local search, which would lead to inaccurate results and a slow convergence rate [15, 16].

Firefly algorithm (FA) is a new nature-inspired optimization algorithm proposed by Yang [17]. Many researches have proved the efficiency of firefly algorithm in different applications, such as electricity price forecasting [18], manufacturing cell formation [19], economic dispatch problem [20], and image analysis [21–23]. In particular, firefly algorithm has been successfully applied to solving multilevel image segmentation problem [24–27]. However, in some cases, the firefly algorithm is easily trapped into local optima [28, 29] which causes premature problem and time-consuming.

Thus, this paper presents an improved firefly algorithm (IFA) to search multilevel thresholds. In IFA, a diversity enhancing strategy with Cauchy mutation is proposed to increase the global search ability. Moreover, a neighborhood strategy is present for local search. The main idea of IFA is adaptively choosing one strategy to help fireflies find the optima according to different stagnation stations. Furthermore, the multilevel Otsu thresholding function is considered as the objective function, and the IFA is applied to searching multilevel thresholds.

The rest of this paper is organized as follows. In Section 2, multilevel image segmentation problem is formulated. Section 3 proposes a novel improved firefly algorithm and describes the application of IFA to multilevel thresholding problem. In Section 4, experiments and comparisons are carried out and the corresponding results are discussed. Section 5 gives a conclusion.

2. Problem Formulation of Multilevel Image Segmentation

With the purpose of separating multiple objects from background, multilevel image segmentation is formulated. Otsu thresholding is a classical and efficient algorithm for image segmentation [30–32]. In consequence, Otsu thresholding is selected to solve image segmentation problem in this paper. The core idea of the Otsu thresholding algorithm is searching a threshold to maximize the between-class variance.

Suppose that there are N pixels with L gray levels in an image; the probability distribution of the gray level i ($i = 0, \ldots, L-1$) can be defined by $p_i = h_i/N$, where $\sum_{i=0}^{L-1} p_i = 1$ and h_i represents the number of pixels with the specific gray level i. Hence, the mean value of the total image is $\mu_T = \sum_{i=0}^{L-1} ip_i$. Let a threshold t partition the image into two classes: class C_1 including the pixels $i \leq t$ and class C_2 including the pixels $i > t$. Define the probability of C_1 and C_2 to be $\omega_1 = \sum_{i=0}^{t} p_i$ and $\omega_2 = \sum_{i=t+1}^{L-1} p_i$. Then, mean values of the two classes can be calculated as

$$\mu_1 = \sum_{i=0}^{t} \frac{ip_i}{\omega_1}, \qquad \mu_2 = \sum_{i=t+1}^{L-1} \frac{ip_i}{\omega_2}. \tag{1}$$

In this situation, the maximum variance between two classes can be defined as

$$\sigma^2 = \omega_1 (\mu_1 - \mu_T)^2 + \omega_2 (\mu_2 - \mu_T)^2. \tag{2}$$

To solve the multilevel Otsu thresholding problem, an image needs to be classified into j classes $(C_1, C_2, \ldots, C_j)$ with the set of thresholds $(t_1, t_2, \ldots, t_{j-1})$. In a similar way, the maximum between-class variance with multilevel thresholds can be defined as

$$\sigma_{\text{mul}}^2 = \sum_{k=1}^{j} \omega_k (\mu_k - \mu_T)^2. \tag{3}$$

with

$$\begin{aligned} &\mu_k = \sum_{i=0}^{t_k} \frac{ip_i}{\omega_k}, \\ &\omega_k = \sum_{i=0}^{t_k} p_i, \\ &\qquad k = 1, \\ &\mu_k = \sum_{i=t_{k-1}+1}^{t_k} \frac{ip_i}{\omega_k}, \\ &\omega_k = \sum_{i=t_{k-1}+1}^{t_k} p_i, \\ &\qquad 1 < k < j, \\ &\mu_k = \sum_{i=t_{k-1}+1}^{L-1} \frac{ip_i}{\omega_k}, \\ &\omega_k = \sum_{i=t_{k-1}+1}^{L-1} p_i, \\ &\qquad k = j, \end{aligned} \tag{4}$$

In other words, the problem of multilevel Otsu thresholding can be understood as searching a set of thresholds $(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{j-1})$ that can maximize the between-class variance. The optimization problem is defined as

$$\left(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{j-1}\right) = \max \sigma_{\text{mul}}^2. \tag{5}$$

Obviously, multilevel Otsu thresholding algorithm is time-consuming and involves large computation. To solve this problem, an improved firefly algorithm is applied to searching multilevel thresholds.

3. Multilevel Image Segmentation Based on IFA

3.1. Firefly Algorithm. Firefly algorithm is a new nature-inspired approach for optimization, which mimics the bioluminescent behavior of fireflies [17]. In firefly algorithm, there are three idealized rules: (1) all fireflies are unisex so that one firefly is attracted to other fireflies regardless of their sex; (2) attractiveness is proportional to their brightness; thus, for any two flashing fireflies, the less bright one moves towards the brighter one. If there is no brighter one than a particular firefly, it will move randomly; (3) the brightness of a firefly is affected or determined by the landscape of the objective function.

The movement of firefly i attracted to another more attractive firefly j is determined by

$$x_i^{(t+1)} = x_i^{(t)} + \beta\left(r_{ij}\right) \times \left(x_j^{(t)} - x_i^{(t)}\right) + \alpha \times \left(\text{rand} - \frac{1}{2}\right), \tag{6}$$

$$\beta\left(r_{ij}\right) = \beta_0 e^{-\gamma r_{ij}{}^2}, \tag{7}$$

where $x_i^{(t)}$ and $x_j^{(t)}$ represent the solution of the tth iteration for firefly i and firefly j; $\beta(r_{ij})$ represents the attraction of firefly j to firefly i, where r_{ij} is the distance of firefly i and firefly j, β_0 is the attractiveness at $r_{ij} = 0$, and γ is the light absorption coefficient. The third term is disturbing term with α being the randomization parameter. rand is a random number generator uniformly distributed in $[0, 1]$.

The main steps of firefly algorithm can be concluded as follows: (1) calculate the fitness values of fireflies according to the objective function; (2) the fireflies with lower fitness values move to ones with higher fitness values according to (6); (3) sort the fireflies and find the one with highest fitness value; (4) repeat iterations until the maximum iteration number is reached.

3.2. Improved Firefly Algorithm. In the previous section, the original firefly algorithm has been introduced. Many researchers have proven the efficiency of firefly algorithm in solving optimization problems. Nevertheless, in some cases, the algorithm is easy to fall into local optima which may lead to inappropriate results and slow convergence. The reason causing the above phenomena is the imbalance of global search and local search. Hence, to solve the problem, two strategies (i.e., diversity enhancing strategy with Cauchy mutation and neighborhood searching strategy) are proposed in Sections 3.2.1 and 3.2.2. A novel improved firefly algorithm with the two strategies is put forward in Section 3.2.3.

3.2.1. Diversity Enhancing Strategy with Cauchy Mutation. In order to increase the global search ability and prevent the loss of diversities, a diversity enhancing strategy with Cauchy mutation is proposed.

The probability density function of one-dimensional Cauchy distribution can be described by

$$f(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2}, \tag{8}$$

where $t > 0$ is a scale parameter [33]. The distribution function is

$$F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{t}\right). \tag{9}$$

In this paper, diversity enhancing mechanism with Cauchy mutation is defined by

$$x_i' = x_i + \text{Cauchy}(0, \delta), \tag{10}$$

where Cauchy$(0, \delta)$ denotes a random number with Cauchy mutation and δ is the scale parameter. Moreover, this strategy is not generated for each individual, but for a certain number of random chosen individuals. Figure 1 illustrates the diversity enhancing process with Cauchy mutation. The main steps of diversity enhancing strategy with Cauchy mutation are as follows: (1) randomly select the S_1 number of fireflies; (2) update the fireflies with (10). It is worth noting that the firefly with current best fitness value is not to be chosen to update with Cauchy mutation. The purpose is to avoid the retrogression phenomenon in the optimum search progress.

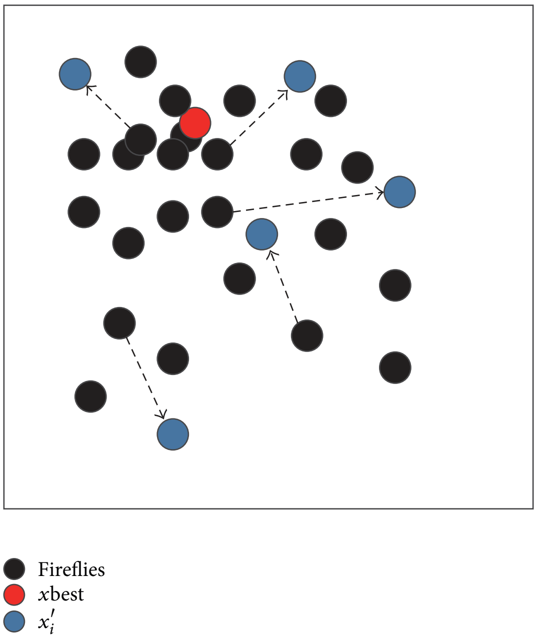


Figure 1: Diversity enhancing strategy with Cauchy mutation.

3.2.2. Neighborhood Strategy. The diversity enhancing strategy with Cauchy mutation increases the global search ability. Nevertheless, an excellent optimization algorithm should keep the balance of exploration and exploitation. Hence, a neighborhood strategy for local search is proposed in this section. The concept of neighborhood in this paper is not the neighborhood individuals in the solution space [34, 35], but the neighborhood values of the firefly with best solution.

Suppose $x\text{best}$ is the best solution of each iteration; $x\text{best}_k$ denotes the kth value in $x\text{best}$. The neighborhood of $x\text{best}$ is defined as

$$\text{Nei}_k = \left[x\text{best}_k - md \cdots x\text{best}_k - d x\text{best}_k x\text{best}_k + d \cdots x\text{best}_k + md\right]', \tag{11}$$

where d denotes the adjacent distance and m is a scale parameter. Therefore, a new firefly x_i' generated by neighborhood strategy is

$$x_i'(k) = x\text{best}_k + \text{rand} \times d, \tag{12}$$

where rand is a random value selected from $[-m, m]$. That is to say, in neighborhood strategy, the process of generating a new firefly is as follows: randomly choose one element from the $2m + 1$ neighborhood values of $x\text{best}$ in each dimension and combine the chosen elements to generate a new firefly.

```
(1)  Set the parameters and generate initial fireflies
(2)  Define the objective function
(3)  Light intensity is determined by the objective function value
(4)  While ($t$ < Maxiter)
(5)    Calculate the fitness value of every firefly, and rank the fireflies
(6)    Find the current best solution and the corresponding fitness value $f_{\text{best}}(t)$
(7)    Update fireflies according to (6)
(8)    If $f_{\text{best}}(t) = f_{\text{best}}(t-1)$
(9)        Count continuously occurred number $N_e$
(10)       If $\text{thr}_1 < N_e < \text{thr}_2$
(11)         Executive Diversity enhancing strategy with Cauchy mutation
(12)       Elseif $\text{thr}_2 \le N_e < \text{thr}_3$
(13)         Randomly select one strategy from Diversity enhancing strategy with Cauchy mutation
             and Neighborhood strategy
(14)       Elseif $\text{thr}_3 \le N_e < \text{thr}_4$
(15)         Executive Neighborhood strategy
(16)       Elseif $N_e \ge \text{thr}_4$
(17)         Terminate and go Step (24)
(18)       End If
(19)       $N_e = 0$
(20)    Elseif $(f_{\text{best}}(t) - f_{\text{best}}(t-1)) < \text{thr}_5$
(21)       Randomly select one strategy from Diversity enhancing strategy with Cauchy mutation
           and Neighborhood strategy
(22)    End If
(23)  End While
(24)  Output the optimum result
```

Algorithm 1

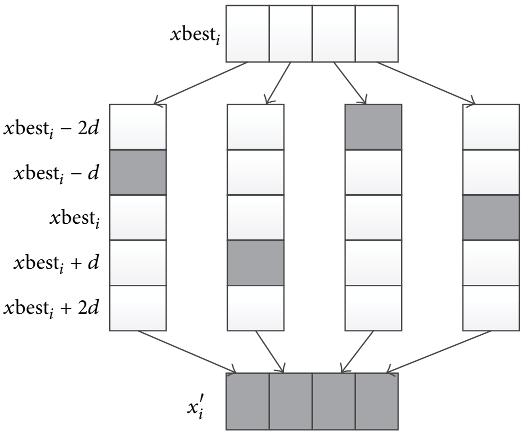


Figure 2: The process of generating a new firefly by neighborhood strategy.

A clear explanation of the process is presented in Figure 2. In order to punish the worst fireflies and accelerate the convergence, the neighborhood strategy is not generated for each firefly, but for a certain number of ranked bottom fireflies. Therefore, the neighborhood strategy can be described briefly as follows: choose the S_2 number of ranked bottom fireflies and update the fireflies according to (12).

3.2.3. Improved Firefly Algorithm. The purpose of the proposed improved firefly algorithm is to help fireflies escape from local optima and accelerate the convergence. The main steps of IFA are described in Algorithm 1, where t is the iteration number, Maxiter is the maximum number of iterations, $f_{\text{best}}(t-1)$ indicates the best fitness value of last iteration, and $f_{\text{best}}(t)$ records the current best fitness value. In Algorithm 1, steps (8) to (22) are the essence of the proposed algorithm. When the best fitness value does not improve or shows very small improvement, it is considered that the fireflies may get stuck in local optima.

In case of $f_{\text{best}}(t) = f_{\text{best}}(t-1)$, count the continuously occurring number N_e. According to the value of N_e, the stagnated situation is divided into four stages with four thresholds (thr_1, thr_2, thr_3, and thr_4). In the first stagnated stage ($\text{thr}_1 < N_e < \text{thr}_2$), the proposed diversity enhancing strategy with Cauchy mutation is applied to increasing the diversity and enhancing the global search ability. In the second stagnated stage ($\text{thr}_2 \le N_e < \text{thr}_3$), two situations are likely to occur: fireflies get stuck in a local optimum that may be nearby or far from the real optimum. As a result, the Cauchy mutation strategy and the neighborhood strategy are selected randomly. If there is no improvement after the second stage ($\text{thr}_3 \le N_e < \text{thr}_4$), it is considered that the local optimum is infinitely close to the real optimum. Therefore, the neighborhood strategy is applied to enhancing the local search ability. When the continuously occurring number N_e achieves the threshold thr_4, it is considered that the global optimum has been found after the earlier three stagnated situations and the iteration would be terminated. In addition, if the difference between $f_{\text{best}}(t)$ and $f_{\text{best}}(t-1)$ is less than the threshold thr_5, the situation is similar to the second stagnated stage. Hence, the approach of this situation is the same as that of the second stage.

Table 1: Parameters of IFA.

Parameters	Values
Number of fireflies	50
Maximum number of iterations	100
Light absorption coefficient γ	1
Initial attractiveness β_0	1
Initial randomization parameter α	1
Threshold values of thr_1, thr_2, thr_3, and thr_4	0, 6, 19, 25
Threshold value of thr_5	0.25
Scale parameter δ in Cauchy mutation	2
Number of chosen fireflies in two strategies S_1 and S_2	5, 5
Scale parameter m in neighborhood strategy	2
Adjacent distance d in neighborhood strategy	1

Parameter α in original firefly algorithm is a randomization parameter. Normally, a large α encourages global search while a small α facilitates the local search [24]. As a result, many researchers set parameter α as a monotonically decreasing function with iteration. However, in the situation of trapping in a local optimum, the small α may not play a role. Therefore, parameter α in IFA is defined as

$$\alpha_t = \begin{cases} \left(\dfrac{1}{2\text{Maxiter}}\right)^{1/(2\text{Maxiter})} \alpha_{t-1}, & \text{if } f_{\text{best}}(t) = f_{\text{best}}(t-1) \\ & \text{or } 0 < \left(f_{\text{best}}(t) - f_{\text{best}}(t-1)\right) < \text{thr}_5 \\ \left(\dfrac{1}{2\text{Maxiter}}\right)^{1/\text{Maxiter}} \alpha_{t-1}, & \text{otherwise.} \end{cases} \tag{13}$$

From (13), in the case of $f_{\text{best}}(t) = f_{\text{best}}(t-1)$ and $(f_{\text{best}}(t) - f_{\text{best}}(t-1)) < \text{thr}_5$, the value of α reduces the speed of recession, so as to increase the disturbance ability.

3.3. Multilevel Image Segmentation Based on IFA. In this section, multilevel image segmentation based on IFA is proposed. As mentioned above, the multilevel image segmentation can be considered as an optimum problem. Therefore, the main steps of the proposed method are as follows:

(1) Input image.

(2) Initialize the parameters of IFA and make the maximum between-class variance σ^2_{mul} be the objective function of IFA.

(3) Use IFA to obtain a set of thresholds $(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{j-1})$, which can maximize (5).

(4) Segment image with thresholds $(\hat{t}_1, \hat{t}_2, \ldots, \hat{t}_{j-1})$.

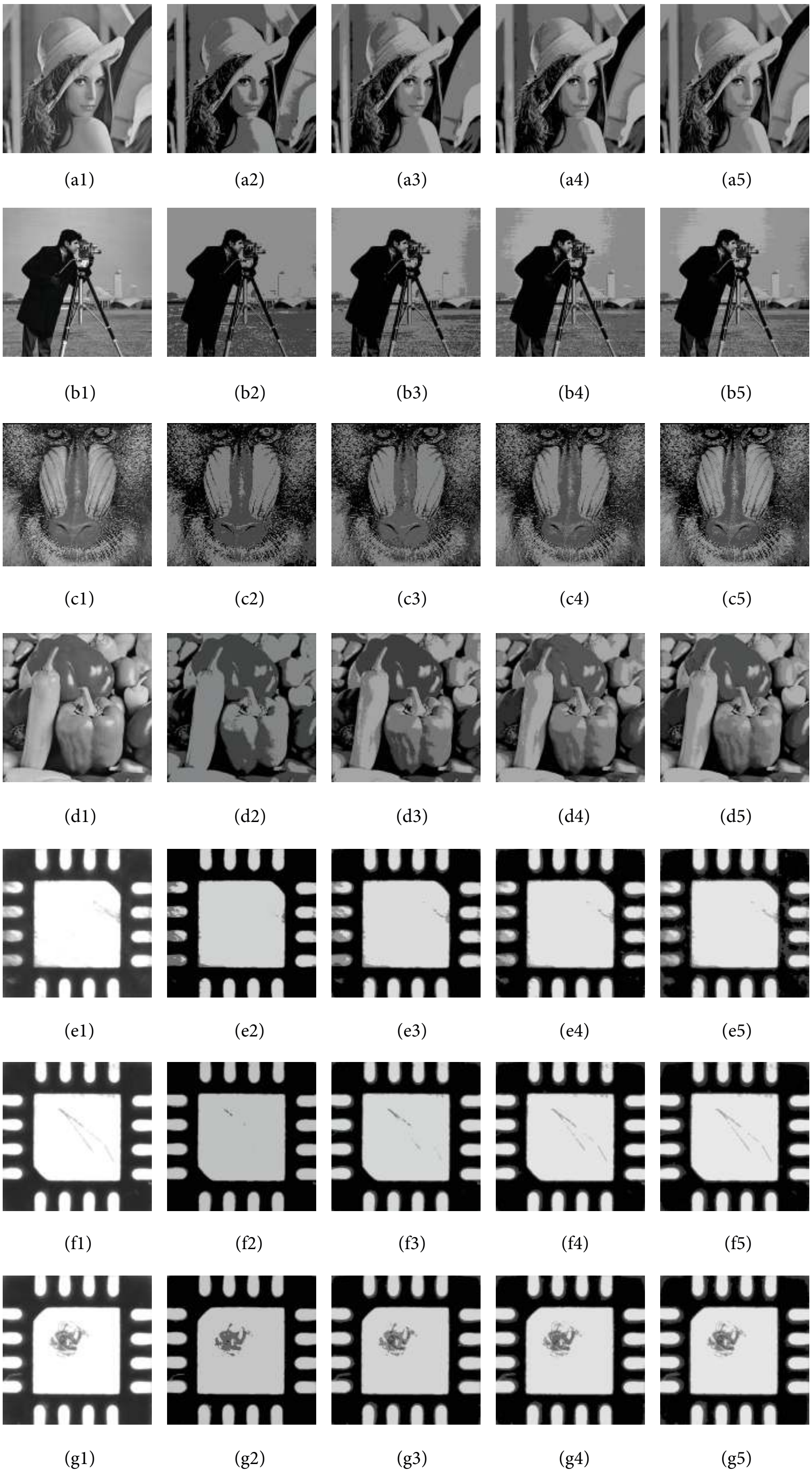
4. Experiment Results and Comparisons

The experiments were implemented in MATLAB on a computer with Intel Core 2.26 GHz and 2 GB memory. Four standard test images ("Lena," "Cameraman," "Baboon," and "Peppers") are used for conducting experiments. Three QFN defect images (QFN images with scratch defect, scrape defect, and void defect) captured from QFN test handler were also used for testing. In order to verify the efficiency of the proposed method, Darwinian particle swarm optimization (DPSO) [10], hybrid differential evolution optimization (HDE) [36], and FA are carried out for comparison.

4.1. Image Segmentation Results of the Proposed Method. According to Section 3 and the characteristics of image segmentation, the parameters of IFA were set as shown in Table 1.

Image segmentation results of the proposed method are shown in Figure 3: from top to bottom, they are "Lena," "Cameraman," "Baboon," "Peppers," "QFN image with scratch defect (hereinafter referred to as QFN1)," "QFN image with scrape defect (hereinafter referred to as QFN2)," and "QFN image with void defect (hereinafter referred to as QFN3)"; the first column shows the original images and the second to the fifth columns show, respectively, the corresponding images of segmentation results with two, three, four, and five thresholds obtained by the proposed method.

4.2. Results Comparisons. The efficiency of the proposed method was evaluated by comparing the results with DPSO, HDE, and FA. DPSO [10] is the variation of particle swarm optimization. In DPSO, there are many swarms existing in each iteration. Moreover, in order to escape the local optima, the swarms are judged according to the selection mechanism. HDE [36] is the variation of differential evolution optimization. In order to improve convergence, the local search routine and adaptive crossover operator are applied in HDE.


(a1) (a2) (a3) (a4) (a5)

(b1) (b2) (b3) (b4) (b5)

(c1) (c2) (c3) (c4) (c5)

(d1) (d2) (d3) (d4) (d5)

(e1) (e2) (e3) (e4) (e5)

(f1) (f2) (f3) (f4) (f5)

(g1) (g2) (g3) (g4) (g5)

Figure 3: Original test images and segmented images with 2, 3, 4, and 5 thresholds.

Table 2: Parameters of DPSO, HDE, and FA.

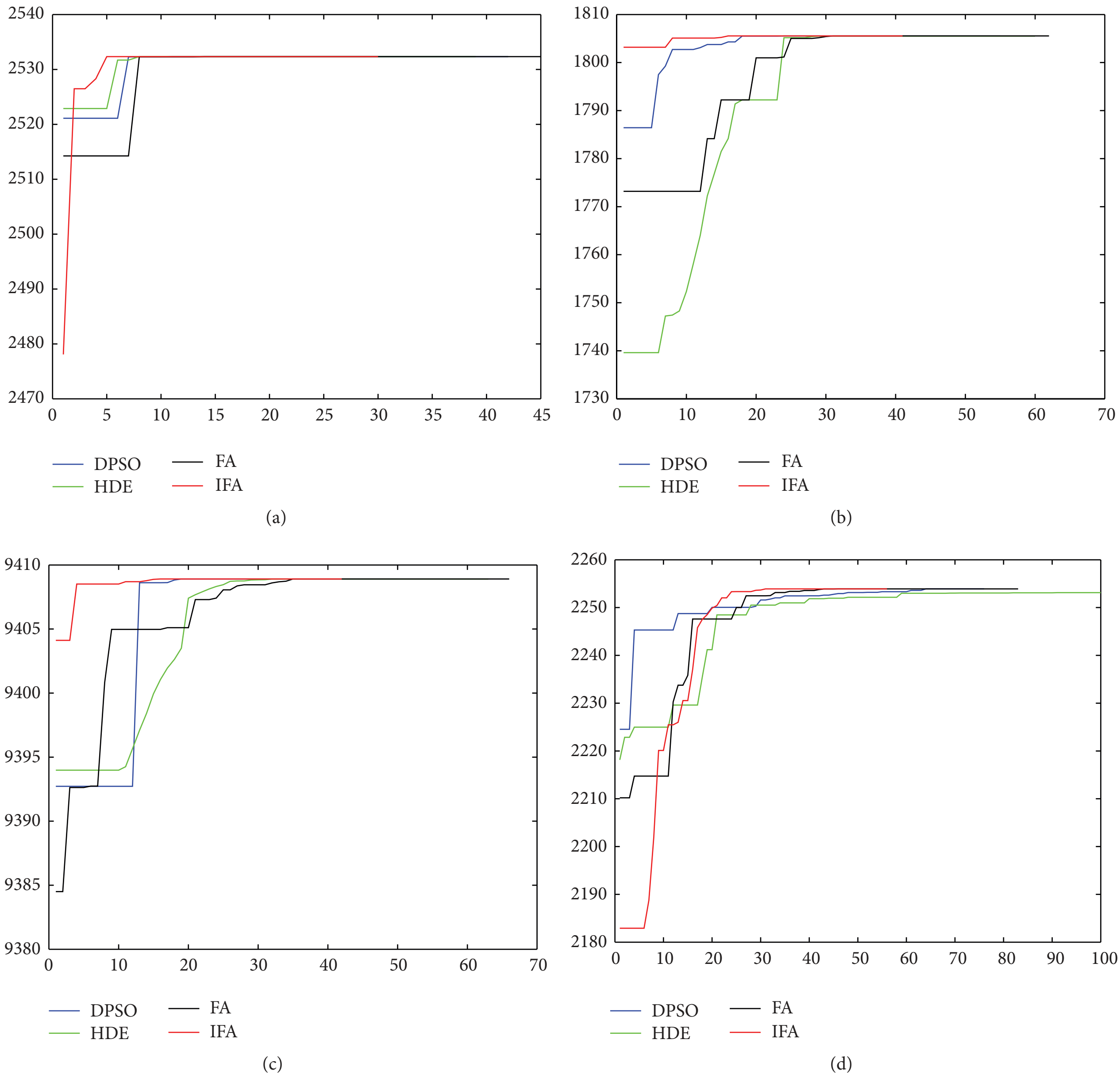
DPSO		HDE		FA	
Population size	15	Population size	50	Number of fireflies	50
Maximum number of iterations	100	Maximum number of iterations	100	Maximum number of iterations	100
Cognitive coefficient	1.5	Triangular distribution for scaling factor	[0.3 0.4 0.5]	Light absorption coefficient	1
Velocities range	[−2, 2]	Triangular distribution for crossover rate	[0.2 0.5 1.0]	Initial attractiveness	1
Initial weight	1.2	Lineal recombination factor	0.75		
Max. population size	30	Refreshment size	20		
Min. population size	5	Cumulative number of success	15		
Number of swarms	4	Probability to run a local search procedure	0.001		
Maximum swarms	6				
Minimum swarms	2				
Stagnancy	10				

Table 3: Threshold values obtained by the four methods.

Test images	K	DPSO	HDE	FA	IFA
Lena	2	87, 146	87, 146	87, 146	87, 146
	3	76, 122, 166	76, 122, 166	76, 122, 166	76, 122, 166
	4	72, 111, 142, 177	72, 111, 142, 177	72, 111, 142, 177	72, 111, 142, 177
	5	61, 88, 115, 142, 177	60, 87, 113, 143, 177	60, 88, 115, 142, 176	61, 88, 115, 142, 177
Cameraman	2	69, 143	69, 143	69, 143	69, 143
	3	58, 118, 155	58, 118, 155	58, 118, 155	58, 118, 155
	4	41, 93, 138, 168	41, 94, 139, 169	41, 94, 139, 169	41, 94, 139, 169
	5	34, 81, 121, 148, 172	33, 79, 119, 147, 171	35, 81, 121, 148, 172	35, 81, 121, 148, 172
Baboon	2	72, 126	72, 126	72, 126	72, 126
	3	55, 92, 133	55, 92, 133	55, 92, 134	55, 92, 133
	4	46, 80, 109, 142	45, 77, 108, 141	46, 79, 109, 142	46, 80, 109, 142
	5	45, 77, 105, 132, 156	44, 75, 102, 127, 155	46, 77, 105, 131, 157	46, 77, 105, 132, 157
Peppers	2	67, 134	67, 134	67, 134	67, 134
	3	62, 118, 165	61, 117, 165	62, 118, 165	62, 118, 165
	4	46, 85, 125, 168	46, 85, 125, 168	46, 85, 125, 168	46, 85, 125, 168
	5	42, 78, 112, 145, 176	42, 78, 112, 145, 177	42, 78, 112, 145, 176	42, 78, 112, 145, 176
QFN1	2	113, 209	113, 209	113, 209	113, 209
	3	80, 145, 220	80, 145, 220	80, 145, 220	80, 145, 220
	4	70, 114, 172, 227	70, 114, 172, 227	70, 114, 172, 227	70, 114, 172, 227
	5	62, 88, 132, 182, 230	63, 91, 136, 186, 231	62, 88, 132, 183, 230	62, 88, 132, 182, 230
QFN2	2	91, 193	91, 193	91, 193	91, 193
	3	68, 133, 215	68, 133, 215	68, 133, 215	68, 133, 215
	4	60, 103, 168, 229	60, 103, 168, 229	60, 103, 168, 229	60, 103, 168, 229
	5	53, 79, 123, 180, 222	54, 80, 124, 181, 232	54, 81, 125, 182, 233	54, 81, 125, 182, 233
QFN3	2	98, 198	98, 198	98, 198	98, 198
	3	72, 133, 212	72, 133, 212	72, 133, 212	72, 133, 212
	4	64, 107, 165, 225	64, 107, 165, 224	64, 107, 165, 225	64, 107, 165, 225
	5	57, 86, 128, 179, 229	58, 86, 129, 180, 230	58, 86, 129, 180, 230	58, 86, 129, 180, 230

The four methods were compared in four aspects: accuracy, computational time, convergence, and stability. Parameters of DPSO, HDE, and FA were set in Table 2. For fair comparison, in the iteration, if global best solution was not improved and the repeat time reached the threshold thr_4 in IFA, DPSO, HDE, and FA would be terminated and output the results. Owing to the randomness of the four algorithms, each algorithm was executed 50 times to verify the performance.

4.2.1. Comparison of Accuracy. Due to the randomness, the best experimental results obtained by the four algorithms are listed in Tables 3 and 4, where K represents the number of thresholds. Tables 3 and 4, respectively, show the thresholds



(a)

(b)

(c)

(d)

Figure 4: Convergence curves.

and the corresponding fitness values. The best values of the four algorithms are emphasized in boldface. It can be found that the four algorithms obtain the best fitness values when $K = 2, 3$. However, with the number of thresholds increasing, the fitness values of IFA are slightly higher than those of the other three methods.

4.2.2. Comparison of Computational Time. The minimum computational time spent on finding the best fitness values in Table 4 is listed in Table 5. The results of computational time reveal that, in a few cases, the DPSO method and HDE method cost less time. However, the IFA method spends the minimum time in most situations.

4.2.3. Comparison of Convergence. In order to demonstrate the convergence performance of the proposed method, convergence curves are drawn in this section. The convergence curves for Peppers ($K = 2$), Baboon ($K = 3$), QFN3 ($K = 4$), and Lena ($K = 5$) are, respectively, shown in Figures 4(a), 4(b), 4(c), and 4(d). It is easy to find that the IFA based method has a faster convergence speed than the other methods. The DPSO method and the HDE method rank second and third. The FA method has the slowest convergence speed.

4.2.4. Comparison of Stability. As mentioned before, all the optimization algorithms are of the randomness characteristic. Tables 3, 4, and 5 show the best results of the four algorithms.

Table 4: Objective function values obtained by the four methods.

Test images	K	Objective function values			
		DPSO	HDE	FA	IFA
Lena	2	**1993.294**	**1993.294**	**1993.294**	**1993.294**
	3	**2162.980**	**2162.980**	**2162.980**	**2162.980**
	4	**2229.253**	**2229.253**	**2229.253**	**2229.253**
	5	**2253.910**	2253.905	**2253.910**	**2253.910**
Cameraman	2	**3650.335**	**3650.335**	**3650.335**	**3650.335**
	3	**3725.715**	**3725.715**	**3725.715**	**3725.715**
	4	3780.663	**3780.687**	**3780.687**	**3780.687**
	5	3811.995	3811.969	**3812.009**	**3812.009**
Baboon	2	**1712.177**	**1712.177**	**1712.177**	**1712.177**
	3	**1805.536**	**1805.536**	**1805.536**	**1805.536**
	4	**1858.560**	1858.496	**1858.560**	**1858.560**
	5	1888.173	1887.624	**1888.204**	**1888.204**
Peppers	2	**2532.321**	**2532.321**	**2532.321**	**2532.321**
	3	**2703.572**	**2703.572**	**2703.572**	**2703.572**
	4	**2766.459**	**2766.459**	**2766.459**	**2766.459**
	5	**2810.842**	2810.830	**2810.842**	**2810.842**
QFN1	2	**8639.842**	**8639.842**	**8639.842**	**8639.842**
	3	**8712.686**	**8712.686**	**8712.686**	**8712.686**
	4	**8744.161**	**8744.161**	**8744.161**	**8744.161**
	5	**8759.376**	8759.372	8759.375	**8759.376**
QFN2	2	**9863.972**	**9863.972**	**9863.972**	**9863.972**
	3	**9925.391**	**9925.391**	**9925.391**	**9925.391**
	4	**9953.834**	**9953.834**	**9953.834**	**9953.834**
	5	9967.853	9967.858	**9967.862**	**9967.862**
QFN3	2	**9303.764**	**9303.764**	**9303.764**	**9303.764**
	3	**9376.492**	**9376.492**	**9376.492**	**9376.492**
	4	**9408.903**	9408.896	**9408.903**	**9408.903**
	5	9425.306	**9425.307**	**9425.307**	**9425.307**

However, it is necessary to calculate the mean value and standard deviation so as to show the quality of stability. The mean value μ and standard deviation std are, respectively, defined as

$$\mu = \frac{\sum_{i=1}^{n_r} F_i}{n_r},$$
$$\text{std} = \sqrt{\frac{\sum_{i=1}^{n_r} \left(F_i - \mu\right)^2}{n_r}}, \quad (14)$$

where n_r is the number of times the algorithm ran and F is the best objective value obtained each time. The mean values and standard deviations of four algorithms are shown in Table 6. As can be seen, the four methods almost get the best stability when the number of thresholds is 2. With the number of thresholds increasing, the IFA algorithm owns the best stability quality.

Table 5: Computational time of the four methods.

Test images	K	Computational time (s)			
		DPSO	HDE	FA	IFA
Lena	2	0.606	0.599	0.719	**0.571**
	3	0.865	**0.734**	1.312	0.761
	4	1.241	1.562	1.438	**1.107**
	5	1.566	1.799	1.753	**1.327**
Cameraman	2	0.633	0.704	0.761	**0.548**
	3	0.969	1.027	1.346	**0.772**
	4	1.337	1.302	1.592	**1.134**
	5	1.945	1.699	1.728	**1.361**
Baboon	2	0.599	**0.513**	0.799	0.574
	3	0.969	1.057	1.222	**0.874**
	4	1.292	1.322	1.377	**0.989**
	5	1.556	1.447	1.665	**1.286**
Peppers	2	0.677	0.628	0.816	**0.609**
	3	0.929	0.816	1.105	**0.805**
	4	1.026	0.973	1.389	**0.894**
	5	1.877	1.291	1.509	**1.031**
QFN1	2	0.469	0.448	0.611	**0.439**
	3	**0.718**	0.841	1.099	0.732
	4	1.021	1.157	1.251	**0.991**
	5	1.343	1.329	2.100	**1.025**
QFN2	2	**0.386**	0.653	0.756	0.531
	3	0.750	0.811	1.018	**0.603**
	4	**0.933**	1.125	1.355	1.037
	5	1.498	1.372	1.867	**1.132**
QFN3	2	**0.487**	0.779	0.881	0.571
	3	0.738	1.115	1.168	**0.669**
	4	1.045	1.503	1.618	**0.850**
	5	**1.075**	1.718	1.778	1.199

4.3. Discussion. In the previous sections, the optimum results of the proposed method and the comparisons of four methods are given. In terms of accuracy, the IFA based thresholding method obtains slightly higher fitness values than the other three methods. On the part of computational time and convergence, IFA, DPSO, and HDE perform better than FA. In the aspect of stability, IFA has the best performance.

The superior performance by IFA is due to the diversity enhancing strategy with Cauchy mutation and neighborhood strategy. Two strategies are applied according to different level of stagnations. With the advance of the two strategies, the IFA based method can keep the balance of exploration and exploitation in the search space. The experimental results indicate that the proposed method performs better than the other three methods.

5. Conclusion

This paper presents a multilevel image thresholding approach based on the proposed improved firefly algorithm. There are two contributions in this paper. Firstly, an improved firefly algorithm is proposed. In IFA, diversity enhancing strategy

Table 6: Stability of the four methods.

Test images	K	DPSO		HDE		FA		IFA	
		μ	std	μ	std	μ	std	μ	std
Lena	2	1993.285	0.067	**1993.294**	**0**	**1993.294**	**0**	**1993.294**	**0**
	3	2162.955	0.098	2162.923	0.371	**2162.980**	**0**	**2162.980**	**0**
	4	2228.640	1.977	2229.134	1.293	2229.132	0.633	**2229.137**	**0.633**
	5	2253.548	0.618	2252.909	2.887	2252.070	3.977	**2253.659**	**0.402**
Cameraman	2	3650.334	0.012	**3650.335**	**0**	**3650.335**	**0**	**3650.335**	**0**
	3	3725.571	0.604	3725.711	0.027	3725.714	3.260*E* − 03	**3725.715**	**0**
	4	3780.562	0.367	3780.643	0.708	3780.640	0.214	**3780.673**	**5.511E − 02**
	5	3811.808	0.531	3810.690	1.072	3810.155	2.903	**3811.754**	**1.353**
Baboon	2	**1712.177**	**0**	**1712.177**	**0**	**1712.177**	**0**	**1712.177**	**0**
	3	1805.266	1.131	1805.381	0.465	1805.494	0.199	**1805.536**	**0**
	4	1858.116	1.985	1858.413	0.842	1858.409	0.196	1858.477	**9.585E − 02**
	5	1887.642	1.801	1885.519	1.726	1885.272	4.185	1887.956	**0.674**
Peppers	2	**2532.321**	**0**	**2532.321**	**0**	**2532.321**	**0**	**2532.321**	**0**
	3	2703.459	0.672	2703.408	0.827	**2703.572**	**0**	2703.571	1.293*E* − 03
	4	2766.232	1.396	2766.137	1.869	2766.417	0.171	**2766.455**	**1.245E − 02**
	5	2810.306	1.777	2808.751	1.802	2810.659	0.296	**2810.840**	**6.023E − 03**
QFN1	2	**8639.842**	**0**	**8639.842**	**0**	**8639.842**	**0**	**8639.842**	**0**
	3	8712.645	0.268	8712.658	3.417*E* − 02	8712.685	4.491*E* − 03	**8712.686**	**0**
	4	8744.050	0.364	8742.127	0.852	8744.149	3.443*E* − 02	**8744.160**	**4.386E − 03**
	5	8759.312	0.147	8759.139	1.386	8759.179	0.247	**8759.340**	**4.855E − 02**
QFN2	2	**9863.972**	**0**	9863.964	3.016*E* − 03	**9863.972**	**0**	**9863.972**	**0**
	3	9925.368	0.115	9925.385	1.547*E* − 02	9925.391	1.841*E* − 03	**9925.391**	**1.428E − 04**
	4	9953.760	0.210	9953.812	0.844	9953.825	1.898*E* − 02	**9953.833**	**2.369E − 03**
	5	9967.785	0.179	9967.803	0.917	9967.682	0.213	**9967.851**	**3.303E − 02**
QFN3	2	**9303.764**	**0**	**9303.764**	**0**	**9303.764**	**0**	**9303.764**	**0**
	3	9376.479	0.050	9376.448	3.091*E* − 02	9376.489	6.243*E* − 03	**9376.491**	**3.053E − 03**
	4	9408.861	0.088	9408.356	0.239	9408.849	0.131	**9408.903**	**2.949E − 03**
	5	9425.208	0.255	9425.117	0.685	9425.128	0.283	**9425.270**	**2.298E − 02**

with Cauchy mutation and neighborhood strategy are conducted to increase the global search ability and accelerate the convergence. Secondly, IFA is applied to searching multilevel global best thresholds. Seven images are used to verify the proposed method. The experimental results reveal that the IFA based method can efficiently that the IFA based method can efficiently search the multilevel thresholds and segment images into background and objects. Moreover, the proposed method shows better performance than the DPSO method, HDE method, and FA method.

Conflict of Interests

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

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (Grant no. 51275090), Science and Technology Support Plan Foundation of Jiangsu (Grant no. BE2008081), and Science and Technology Achievements Transformation Foundation of Jiangsu (Grant no. BA2010093).

References

[1] K. Hanbay and M. F. Talu, "Segmentation of SAR images using improved artificial bee colony algorithm and neutrosophic set," *Applied Soft Computing Journal*, vol. 21, pp. 433–443, 2014.

[2] J. Li, X. Rao, F. Wang, W. Wu, and Y. Ying, "Automatic detection of common surface defects on oranges using combined lighting transform and image ratio methods," *Postharvest Biology and Technology*, vol. 82, pp. 59–69, 2013.

[3] J. W. Funck, Y. Zhong, D. A. Butler, C. C. Brunner, and J. B. Forrer, "Image segmentation algorithms applied to wood defect detection," *Computers and Electronics in Agriculture*, vol. 41, no. 1–3, pp. 157–179, 2003.

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

[5] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285–296, pp. 23–27, 1975.

[6] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, and Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.

[7] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[8] A. Abutaleb and A. Eloteifi, "Automatic thresholding of gray-level pictures using 2-D entropy," in *Proceedings of the 31st Annual Technical Symposium, Applications of Digital Image Processing X*, vol. 829 of *Proceedings of SPIE*, pp. 29–35, International Society for Optics and Photonics, San Diego, Calif, USA, January 1988.

[9] K. Hammouche, M. Diaf, and P. Siarry, "A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation," *Computer Vision and Image Understanding*, vol. 109, no. 2, pp. 163–175, 2008.

[10] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, "An efficient method for segmentation of images based on fractional calculus and natural selection," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12407–12417, 2012.

[11] Y.-C. Liang, A. H.-L. Chen, and C.-C. Chyu, "Application of a hybrid ant colony optimization for the multilevel thresholding in image processing," in *Neural Information Processing*, vol. 4233 of *Lecture Notes in Computer Science*, pp. 1183–1192, Springer, Berlin, Germany, 2006.

[12] P. D. Sathya and R. Kayalvizhi, "Optimal multilevel thresholding using bacterial foraging algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15549–15564, 2011.

[13] E. Cuevas, D. Zaldivar, and M. Pérez-Cisneros, "A novel multi-threshold segmentation approach based on differential evolution optimization," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5265–5271, 2010.

[14] V. Osuna-Enciso, E. Cuevas, and H. Sossa, "A comparison of nature inspired algorithms for multi-threshold image segmentation," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1213–1219, 2013.

[15] J. Zhang, H. Li, Z. Tang, Q. Lu, X. Zheng, and J. Zhou, "An improved quantum-inspired genetic algorithm for image multilevel thresholding segmentation," *Mathematical Problems in Engineering*, vol. 2014, Article ID 295402, 12 pages, 2014.

[16] J.-Y. Li, Y.-D. Zhao, J.-H. Li, and X.-J. Liu, "Artificial bee colony optimizer with bee-to-bee communication and multipopulation coevolution for multilevel threshold image segmentation," *Mathematical Problems in Engineering*, vol. 2015, Article ID 272947, 23 pages, 2015.

[17] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications*, vol. 5792 of *Lecture Notes in Computer Science*, pp. 169–178, Springer, Berlin, Germany, 2009.

[18] P. Mandal, A. U. Haque, J. Meng, A. K. Srivastava, and R. Martinez, "A novel hybrid approach using wavelet, firefly algorithm, and fuzzy ARTMAP for day-ahead electricity price forecasting," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1041–1051, 2013.

[19] M. K. Sayadi, A. Hafezalkotob, and S. G. J. Naini, "Firefly-inspired algorithm for discrete optimization problems: an application to manufacturing cell formation," *Journal of Manufacturing Systems*, vol. 32, no. 1, pp. 78–84, 2013.

[20] X.-S. Yang, S. S. Sadat Hosseini, and A. H. Gandomi, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect," *Applied Soft Computing*, vol. 12, no. 3, pp. 1180–1186, 2012.

[21] M.-H. Horng, "Vector quantization using the firefly algorithm for image compression," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1078–1091, 2012.

[22] Y. Zhang and L. Wu, "A novel method for rigid image registration based on firefly algorithm," *International Journal of Research and Reviews in Soft and Intelligent Computing*, vol. 2, no. 2, pp. 141–146, 2012.

[23] T. Hassanzadeh, H. Vojodi, and F. Mahmoudi, "Non-linear grayscale image enhancement based on firefly algorithm," in *Swarm, Evolutionary, and Memetic Computing*, pp. 174–181, Springer, Berlin, Germany, 2011.

[24] M.-H. Horng and R.-J. Liou, "Multilevel minimum cross entropy threshold selection based on the firefly algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14805–14811, 2011.

[25] T. Hassanzadeh, H. Vojodi, and A. M. E. Moghadam, "An image segmentation approach based on maximum variance intra-cluster method and Firefly algorithm," in *Proceedings of the 7th*

International Conference on Natural Computation (ICNC '11), vol. 3, pp. 1817–1821, IEEE, Shanghai, China, July 2011.

[26] M.-H. Horng and T.-W. Jiang, "Multilevel image thresholding selection based on the firefly algorithm," in *Proceedings of the 7th International Conference on Ubiquitous Intelligence & Computing and the 7th International Conference on Autonomic & Trusted Computing (UIC/ATC '10)*, pp. 58–63, IEEE, Xian, China, October 2010.

[27] I. Brajevic and M. Tuba, "Cuckoo search and firefly algorithm applied to multilevel image thresholding," in *Cuckoo Search and Firefly Algorithm*, vol. 516 of *Studies in Computational Intelligence*, pp. 115–139, Springer International, New York, NY, USA, 2014.

[28] S. Mohammadi, B. Mozafari, S. Solimani, and T. Niknam, "An Adaptive Modified Firefly Optimisation Algorithm based on Hong's Point Estimate Method to optimal operation management in a microgrid with consideration of uncertainties," *Energy*, vol. 51, pp. 339–348, 2013.

[29] S.-J. Huang, X.-Z. Liu, W.-F. Su, and S.-H. Yang, "Application of hybrid firefly algorithm for sheath loss reduction of underground transmission systems," *IEEE Transactions on Power Delivery*, vol. 28, no. 4, pp. 2085–2092, 2013.

[30] P. Shanmugavadivu and K. Balasubramanian, "Thresholded and optimized histogram equalization for contrast enhancement of images," *Computers & Electrical Engineering*, vol. 40, no. 3, pp. 757–768, 2014.

[31] P. D. Sathya and R. Kayalvizhi, "Optimal segmentation of brain MRI based on adaptive bacterial foraging algorithm," *Neurocomputing*, vol. 74, no. 14-15, pp. 2299–2313, 2011.

[32] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644–1649, 2006.

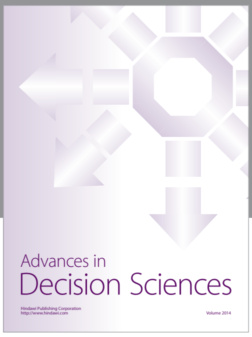
[33] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

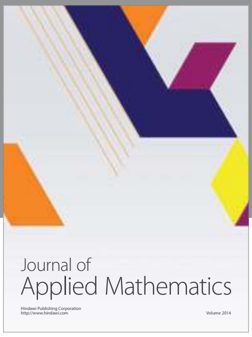
[34] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 526–553, 2009.

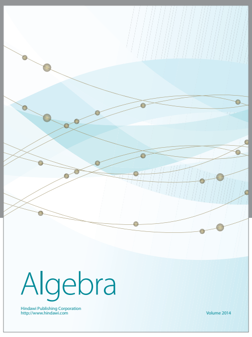
[35] H. Wang, H. Sun, C. Li, S. Rahnamayan, and J.-S. Pan, "Diversity enhanced particle swarm optimization with neighborhood search," *Information Sciences*, vol. 223, pp. 119–135, 2013.

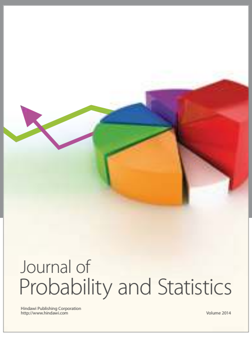
[36] G. Reynoso-Meza, J. Sanchis, X. Blasco, and J. M. Herrero, "Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '11)*, pp. 1551–1556, IEEE, New Orleans, La, USA, June 2011.

Advances in
Operations Research

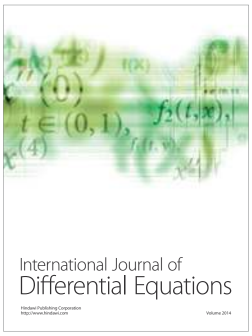

Advances in
Decision Sciences

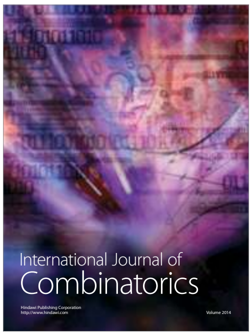

Journal of
Applied Mathematics

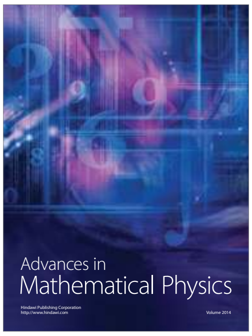

Algebra

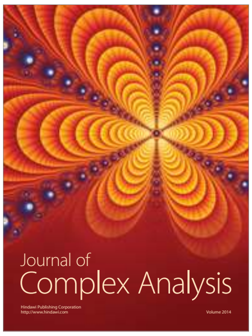

Journal of
Probability and Statistics

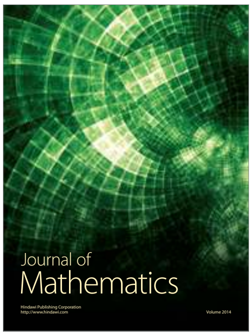
The Scientific
World Journal

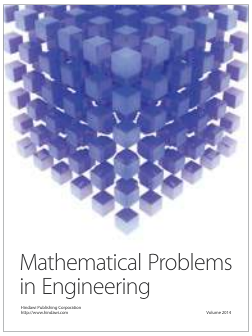

International Journal of
Differential Equations

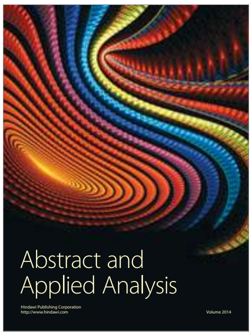

International Journal of
Combinatorics

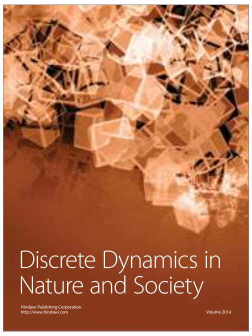

Advances in
Mathematical Physics

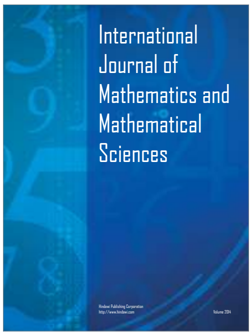

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

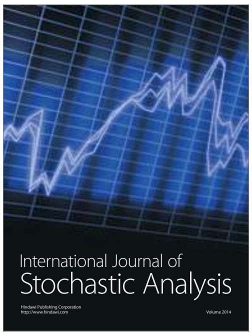

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization