

## Research Article

# A Novel Plant Root Foraging Algorithm for Image Segmentation Problems

Lianbo Ma,<sup>1,2</sup> Kunyuan Hu,<sup>1</sup> Yunlong Zhu,<sup>1</sup> Hanning Chen,<sup>1</sup> and Maowei He<sup>1,2</sup>

<sup>1</sup>Laboratory of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

<sup>2</sup>University of Chinese Academy of Sciences, Beijing 100039, China

Correspondence should be addressed to Yunlong Zhu; 46272589@qq.com and Hanning Chen; 1464071948@qq.com

Received 4 May 2014; Accepted 12 June 2014; Published 16 July 2014

Academic Editor: Haipeng Peng

Copyright © 2014 Lianbo Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new type of biologically-inspired global optimization methodology for image segmentation based on plant root foraging behavior, namely, artificial root foraging algorithm (ARFO). The essential motive of ARFO is to imitate the significant characteristics of plant root foraging behavior including branching, regrowing, and tropisms for constructing a heuristic algorithm for multidimensional and multimodal problems. A mathematical model is firstly designed to abstract various plant root foraging patterns. Then, the basic process of ARFO algorithm derived in the model is described in details. When tested against ten benchmark functions, ARFO shows the superiority to other state-of-the-art algorithms on several benchmark functions. Further, we employed the ARFO algorithm to deal with multilevel threshold image segmentation problem. Experimental results of the new algorithm on a variety of images demonstrated the suitability of the proposed method for solving such problem.

## 1. Introduction

In recent years, biologically-inspired optimization algorithms have been widely applied for solving complex optimization problems. These algorithms have achieved great development and have been proven to be more powerful than the conventional methods based on formal logics or mathematical programming [1–4]. For instances, genetic algorithm (GA), originally proposed by Holland [5], represents a fairly abstract model of Darwinian evolution and biological genetics; ant colony optimization (ACO), conceived by Dorigo et al. [6, 7], is inspired by the ants' social behaviors in finding shortest paths; particle swarm optimization (PSO), proposed by Eberhart and Kennedy [8], gleams ideas from social behavior of bird flocking and fish schooling. However, the existing bionic optimization models usually simulate animal behaviors, and the research of some natural plant-related evolution mechanisms does not obtain enough attention, such as plant root foraging behavior [9]. Currently, capturing the novel properties of potential plants to construct new intelligent computation models has attracted more and more attention.

In plant ecosystem, plant root foraging is viewed as one significant characteristic of plant behavior, which utilizes the morphological and physiological plastic responses affecting individual plant growth [10]. Plastic root foraging responses have been widely recognized as an important strategy for plants to explore heterogeneously distributed resources [11–14]. Root morphological plasticity generates higher elongate-length and branching biomass per unit soil volume in nutrient-rich patches compared to nutrient-poor patches. If a randomly growing root encounters an area high in moisture, it is likely to branch profusely and grow, which is called root tropism. After many generations, the spatial configuration of root system in the soil is shaped with higher root biomass in the area higher in moisture. Logically, such evolutionary principles can be modeled as an optimization process. Here, a novel optimization model based on plant root foraging behaviors is established, called ARFO. This method provides an open framework to utilize research in plant behavioral ecology to tackle complex problems, which set up a new information section mechanism based on auxin to determine how to select new growing points and branching number of roots.

Image segmentation is regarded as a basic operation for meaningful analysis and interpretation of image acquired, responsible for discriminating an object from other objects having distinct gray levels or different colors. As an important and effective tool among the existing segmentation techniques, multilevel threshold segmentation technique can be roughly classified into two types: optimal threshold methods [15–18] and property based threshold methods [19–21]. Those multilevel threshold methods are fast and suitable for the case that the number of thresholds is hard to determine and should be specified in advance. Several algorithms have so far been proposed in literatures that have addressed the issue of optimal threshold [22, 23]. Many of these algorithms attempt to achieve optimal threshold such that the thresholded classes achieve some desired characteristic. References [18, 22, 23] proposed some novel methods derived from optimizing an objective function. However, all of these methods have a common problem, that the computational complexity rises exponentially when extended to multilevel threshold due to the exhaustive search employed, which limits the multilevel threshold applications.

Recently, to address this concerned issue, several swarm intelligence (SI) algorithms as the powerful optimization tools have been introduced to the field of image segmentation owing to their predominant abilities of coping with complex nonlinear optimizations [24, 25]. In this paper, the ARFO algorithm is employed for multilevel threshold image segmentation to prove its effectiveness and robustness, which is based on entropy criterion based fitness measure. Several other mentioned algorithms are tested as a comparison.

The rest of the paper is organized as follows. In Section 2, the root foraging model and the implementation details of ARFO algorithm will be given. Section 3 experimentally compares ARFO with other well-known algorithms on ten benchmark functions. And its application to image segmentation has been presented in Section 4. Finally, Section 5 outlines the conclusion.

## 2. Root Foraging Model for Optimization

*2.1. Root Foraging Model.* The root foraging behavior concept is proposed by McNickle et al. based on MVT theory [9, 26], in which plant roots exhibit morphological responses to environmental heterogeneity through the selective allocation of root biomass per unit time to the nitrogen-rich patch. It is worthy noted that there already exist some effective approaches to mimic such root foraging behaviors, such as  $L$ -system [27–29]. But the key issue for optimization application is not resolved, which is that how to select new growing points to grow and how to ensure the roots towards the optimal position.

In the plant biology, the tropism behaviors of roots system, especially the gravitropism and hydrotropism, are the significantly important characteristics of root behaviors [30]. Gravitropism induces plant roots grow along a specific vector relative to gravity [31] while hydrotropism is the root morphological responses is guided by a stimuli or gradient in nutrition concentration [32]. Note that a kind of essential

hormone information-auxin plays an important role in conducting these tropism behaviors, influencing simultaneously the spatial configuration of the roots system [33]. In order to idealize above plant root foraging behaviors, some criteria are presented as follows.

(1) *Criterion-1: Auxin Regulation.* Auxin conducts the root's adaptive growth and the information exchange among root tips. The roots spatial structure is regulated by the auxin concentration, which is not static, but dynamically reallocated after new roots germinate and grow.

(2) *Criterion-2: Root Growth Behavior.* Regrowing means roots can elongate with a certain angle, and branching is the main progress for multiplying the number of roots, in which new roots originate from the branching of the parent root tip by a certain elongate-unit in new direction.

(3) *Criterion-3: Classification of the Root System.* The whole plant root system is divided into three categories sorted by the auxin distribution: mainroots, the lateral-roots, and the dead-roots. Each root tip of mainroots can implement both regrowing and branching, in which the branching number is determined by auxin concentration. The root tips of lateral-roots can only implement regrowing operation.

(4) *Criterion-4: Root Tropisms.* Root affected by various tropisms evolves towards the most nutrient rich position. Hydrotropism makes the growing direction of the root tips towards the optimal individual position. Meantime gravitropism imposes that roots grow in the direction of gravitational pull (i.e., downward), which means that the growing angle of each root tip is limited to a certain range.

The root foraging model assumes that the optimization problem, whose feasible region is used as soil environment, is a minimization problem with  $D$ -dimensional vectors  $(d^1, d^2, \dots, d^D)$  and a growing point (i.e., root tip) corresponds to a potential solution instance to the problem. Through the regrowing and branching operations, the root tips of the proposed model will quickly spread to the global optimum in the feasible region of optimization problems, until no new branches are generated. Note that, for more precise modeling of the auxin information mechanism, the nutrient concentration strategy is incorporated.

*2.2. Auxin Information.* The auxin information is conducted to regulate the growth points and branching number of roots. We assumed that the sum of auxin concentration  $F_i$  is considered as 1 in the mathematical model of root systems. In order to calculate the  $F_i$  value of every root the next equation is used:

$$f_i = \frac{fit_i - f_{worst}}{f_{best} - f_{worst}}, \quad (1)$$

$$N_i = \frac{Nutition_i - Nutrition_{worst}}{Nutition_{best} - Nutrition_{worst}}, \quad (2)$$

$$F_i = \frac{f_i}{\sum_{j=1}^N f_j} * \xi + \frac{N_i}{\sum_{j=1}^N N_j} (1 - \xi), \quad \xi \in (0, 1), \quad (3)$$

where  $i$  is the position of the growing point,  $\xi$  is a uniform random quantity,  $N$  is the total number of the points,  $fit(\cdot)$  is the fitness value of the point. and  $f_{\text{worst}}$  and  $f_{\text{best}}$  are the maximum and minimum of the current points, respectively.  $Nutrition_i$  is the current nutrient concentration of individual  $i$ .

In addition to the relative position of each point (objective function value), the auxin concentration is also determined by the gradient of function fitness. We establish the concept of nutrient concentration to simulate the moisture gradient of soil environment, expressed as

$$Nutrition_i(t+1) = \begin{cases} Nutrition_i(t) + 1, & \text{if } f_i(t+1) < f_i(t), \\ Nutrition_i(t) - 1, & \text{if } f_i(t+1) > f_i(t). \end{cases} \quad (4)$$

Nutrient concentration reveals that even if the amount of moisture is not best, the higher moisture gradient of the soil is also conducive to the root cell division to absorb water and nutrition.

**2.3. Mainroots Growth Operations.** According to *Criterion-3* of Section 2, at each iteration, the root population consists of three kinds of members: mainroots, lateral-roots, and dead-roots. The root located in the most promising area, conferring the highest auxin concentration value, is chosen as mainroot. It then implements the regrowing operator and branching operator to explore and exploit the whole space. The procedures of simulating foraging behavior of mainroots are abstracted as follows.

**2.3.1. Regrowing Operator.** Since the mainroots with higher auxin concentration values tend to mature, their growth uses larger elongate-length unit, taking on significant hydrotropism and gravitropism characteristics. In order to simulate the influences of these tropisms in the mainroots growth process, the mathematic expression of regrowing operator can be described as follows.

*Step 1.* In each cycle of roots growth process, all root tips are sorted by auxin concentration values. Some excellent root tips with higher auxin concentration values have higher probability to be selected as mainroots for branching operator. The number of the roots chosen as mainroots is defined by

$$S = N * Cr, \quad (5)$$

where  $S$  is the size of mainroots group,  $N$  is the total number of the roots, and  $Cr$  is the selection probability.

*Step 2.* Under hydrotropism, select some of current mainroots to search towards the optimal position of individuals:

$$X_i(t+1) = X_i(t) + R_3(X_{\text{best}} - X_i(t)). \quad (6)$$

*Step 3.* Under gravitropism, new growing points of rest mainroots can be formulated as

$$X_i(t+1) = X_i(t) + R_4 * Grow_{\text{max}} * D_i(\varphi). \quad (7)$$

And the growth angle representation of the mainroot  $i$  is defined by

$$\varphi_i(t+1) = \varphi_i(t) + R_5 * \alpha_{\text{max}} + R_6 * \pi, \quad \alpha_{\text{max}} < \pi, \quad (8)$$

where  $R_3$ ,  $R_5$ , and  $R_6$  are random coefficients varying from 0 to 1,  $R_4$  is a normally distributed random number with mean 0 and standard deviation 1, and  $Grow_{\text{max}}$  is the maximum of elongate-length (i.e., objective function boundary range). To mimic the phenomenon that mainroot tip in the next generation tends to grow downwards under gravitropism, the entire growing angle space  $\alpha_{\text{max}}$  is limited to  $\pi$ .

From (6), the mainroot can rapidly approach a near-optimal solution and stay for some growth cycles in several domains containing local optima. From (7), it can escape from these local optima to explore other domains perhaps containing the global optimum, but it was not able to stop there.

**2.3.2. Branching Operator.** In the root foraging model, the branching operator is the local search strategy to exploit around the optimal area, in which the mainroots branch new roots using a small elongate-length unit, forming a certain density distribution slowly in the promising region's immediate vicinity. This procedure can be described as follows.

*Step 1.* The nutrient concentration ( $Nutrition_i$ ) of mainroot  $i$  is compared with the threshold value  $BranchG$  (satisfied  $0 < BranchG < 1$ ) to determine whether it performs branching operator by

$$\begin{aligned} &\text{branching,} && \text{if } Nutrition_i > BranchG, \\ &\text{nobranching,} && \text{otherelse.} \end{aligned} \quad (9)$$

*Step 2.* Determine the branching number according to the auxin concentration distribution. In principle, the position with higher moisture content and gradient in soil is more conducive to roots branching. The branching number  $w_i$  is calculated as

$$w_i = \begin{cases} S_{\text{min}} F_i * r * (S_{\text{max}} - S_{\text{min}}) + S_{\text{min}}, & \text{if } Nutrition_i > BranchG, \\ 0, & \\ \text{else,} & \end{cases} \quad (10)$$

where  $S_{\text{max}}$  and  $S_{\text{min}}$  are the maximum and minimum of the new growing points, respectively, and  $r$  is random distribution coefficient.  $F_i$  is auxin concentration of mainroot  $i$ .

*Step 3.* Calculate the fitness of new branching points. Regarding the growth direction of the initial mainroot as zero degree, the searching angle space is divided into  $S_{\text{max}}$  subzones and the angle of new growing point is randomly falling within one of  $S_{\text{max}}$  angle subzones. The mathematic equation of new growing point yields

$$X_i(t+1) = X_i(t) + R_1 * D_i(\varphi_i), \quad (11)$$

where  $X_i(t+1)$  is the new growing point from  $X_i(t)$ ,  $R_1$  is the elongate-length unit, which is a random varying from 0 to 1,  $\varphi_i$  is the growth angle  $(\varphi_{i1}, \varphi_{i2}, \dots, \varphi_{i(D-1)})$ , and the growth direction  $D_i(\varphi_i) = (d_{i1}, d_{i2}, \dots, d_{iD})$  which can be calculated from  $\varphi_i$  via a Polar to Cartesian coordinates transformation [34]:

$$\begin{aligned} d_{i1} &= \prod_{k=1}^{D-1} \cos(\varphi_{ik}), \\ d_{ij} &= \sin(\varphi_{ij}) \prod_{k=1}^{D-1} \cos(\varphi_{ik}), \\ d_i^D &= \sin(\varphi_i^{D-1}). \end{aligned} \quad (12)$$

The growth angle  $\varphi_i$  is calculated as follows:

$$\varphi(t+1) = \varphi(t) + R_2 \alpha_{\text{init}} + \frac{K * \beta_{\text{max}}}{S_{\text{max}}}, \quad (13)$$

where  $R_2$  is a random coefficient varying from 0 to 1,  $\alpha_{\text{init}}$  is original growth angle value of the initial mainroot as zero degree,  $K$  is randomly parameter selecting the subzone,  $S_{\text{max}}$  is subzones number, and  $\beta_{\text{max}}$  is the maximum growing turning angle. Similar to (8), considering gravitropism, the growing angle space  $\beta_{\text{max}}$  is limited to  $\pi$  to make mainroots tend to grow downwards.

**2.4. Lateral-Roots Growth Operation.** Due to the fact that the lateral-roots group with lower auxin concentration stays in the initial growth phase, its elongation step is relatively small and the influence of the hydrotropism can be ignored. The lateral-roots with the smaller elongation step are exploiting a local optimal domain, which closed to its original point.

The mathematic expression, which simulates above behavior, can be defined as (14):

$$X_i(t+1) = X_i(t) + R_5 * \text{Grow}_{\text{max}} * D_i(\varphi_i), \quad (14)$$

$$\varphi_i(t+1) = \varphi_i(t) + R_6 \alpha_{\text{max}}, \quad (15)$$

where  $R_5$  is normal distribution function (mean is 0, deviation is 1) and  $R_6$  is a random coefficient.

**2.5. Dead-Roots Elimination.** In the proposed root foraging model, it is assumed that  $N_i$  is the current population size,  $N_i$  will increase by one if a root tip splits and reduce by one if a root dies determined by auxin distribution, and it will vary in the searching process [35]. The branching criterion and dead-roots eliminating criterion are listed in

$$\begin{aligned} N_i &= N_i + w_i, & \text{if } F_i > \text{BranchG}, \\ N_i &= N_i - 1, & \text{if } F_i < N\text{mority}, \end{aligned} \quad (16)$$

where  $w_i$  is the branching number defined by (10), BranchG is the branching threshold value defined by (9), and Nmority is the death threshold value parameter.

**2.6. ARFO Algorithm.** We implement an instantiation algorithm of the proposed root foraging model, namely, artificial root foraging optimizer (ARFO), for numerical function optimization. The pseudocode of the detailed procedure is given as follows.

*Step 1* (initialization). Initialize the positions of root tips population and evaluate the nutrient (fitness) of the population. Set maximum number of cycles (LimitC).

*Step 2.* Iteration = 0.

*Step 3* (auxin concentration calculation). (1) Calculate the nutrient concentration and auxin concentration values of the population by (2) and (3).

(2) Divide the population into two groups: mainroots group and lateral-roots group by (5).

*Step 4* (mainroots operation). Loop over each mainroot tip

- (1) Select a random portion of mainroots group to take regrowing operator by (6), (7), and (8).
- (2) Evaluate the fitness of the renewal mainroots and apply greedy selection.
- (3) Adjust the corresponding nutrient concentration value by (4).
- (4) If the condition of branching determined by (9) is met, continue; otherwise, go to Step 5.
- (5) Calculate the branching number by (10) and the branching new roots by (11).
- (6) The population size increases by the corresponding number calculated by (10).

*Step 5* (lateral-roots operation). Loop over each root tip of lateral-roots

- (1) Lateral-root takes regrowing operator by (14) and (15).
- (2) Evaluate the fitness of the renewal lateral-root and apply greedy selection.
- (3) Adjust the corresponding nutrient concentration value by (4).

*Step 6* (dead roots eliminating). Remove the dead individuals from the population according to their auxin concentration values by (16).

*Step 7.* Iteration = Iteration + 1;

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

*Step 9.* Output the best solution achieved.

By using the regrowing and branching operators, ARFO not only alleviates the premature convergence in the entire solution space, but also improves the accuracy of optimum in the local solution space, at least to some extent.



- (1) *Exploring strategy*. The regrowing operator implemented by mainroots employs a large elongate-length unit to explore the previously unscanned regions in the search space.
- (2) *Exploiting strategy*. To perform fine-tuning exploitation of the global optimum, both the branching operator and the lateral-root's regrowing operator employ a small elongate-length unit and a random angle.

### 3. Benchmark Test

*3.1. Test Functions.* In the benchmark testing studies, according to the no free lunch (NFL) theorem [36], a set of 10 benchmark functions (consisting of basic benchmark functions  $f_1 \sim f_5$  and CEC'05 functions  $f_6 \sim f_{10}$ ) was employed, which are listed as follows. The dimensions, initialization ranges, global optimum, and the criterion of each function are summarized in Table 1.  $f_1 \sim f_5$  were adopted widely in evolutionary computation domain to show solution quality and convergence rate.  $f_6 \sim f_{10}$  are shifted and rotated functions selected from CEC2005 functions; their global optimum values are different to each other. In order to compare the different algorithms fairly, we decide to use the number of function evaluations (FEs) as a time measure substitute the number of iterations.

(1) *Sphere Function*

$$f_1(x) = \sum_{i=1}^n x_i^2. \quad (17)$$

(2) *Rosenbrock Function*

$$f_2(x) = \sum_{i=1}^n 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2. \quad (18)$$

(3) *Rastrigin Function*

$$f_3(x) = \sum_{i=1}^n 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2. \quad (19)$$

(4) *Schwefel Function*

$$f_4(x) = D * 418.9829 + \sum_{i=1}^D -x_i \sin\left(\sqrt{|x_i|}\right). \quad (20)$$

(5) *Griewank Function*

$$f_5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \quad (21)$$

(6) *Shifted Sphere Function*

$$f_6(x) = \sum_{i=1}^D z_i^2 + f_{\text{bias}1}, \quad z = x - o. \quad (22)$$

(7) *Shifted Rosenbrock's Function*

$$f_7(x) = \sum_{i=1}^{D-1} \left(100(z_i^2 - z_{i+1})^2 + (z_i^2 - 1)^2\right) + f_{\text{bias}}. \quad (23)$$

(8) *Shifted Schwefel's Problem 1.2*

$$f_8(x) = \sum_{i=1}^D \left(\sum_{j=1}^i z_j\right)^2 + f_{\text{bias}}, \quad z = x - o. \quad (24)$$

(9) *Shifted Rotated Griewank's Function without Bounds*

$$f_9(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f_{\text{bias}}. \quad (25)$$

(10) *Shifted Rastrigin's Function*

$$f_{10}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{\text{bias}}, \quad z = x - o. \quad (26)$$

*3.2. Parameters Settings for the Involved Algorithms.* Experiments were conducted to compare the performance of ARFO with four successful evolutionary algorithms (EAs) on ten benchmark functions with 20 dimensions and 100 dimensions: standard particle swarm optimization (PSO) [8], real-coded genetic algorithm (GA) [5], differential evolution algorithm (DE) [3], and artificial bee colony algorithm (ABC) [4].

The experiments were run 30 times, respectively, for each algorithm on each benchmark function and the maximum evaluation number is set as 100000. The mean values and standard deviation of the results are presented. Population size in PSO, GA, DE, and ABC was 20. The PSO algorithm we used is the standard one (i.e., the global version with inertia weight). The parameters were given by the default setting of [8]; the acceleration factors  $c_1$  and  $c_2$  were both 2.0; a decaying inertia weight  $\omega$  starting at 0.9 and ending at 0.4 was used. For DE,  $F$  was set to 0.5 and crossover rate was chosen to be 0.9 as recommended in [3]. The GA algorithm we executed is a real-coded genetic algorithm with intermediate crossover and Gaussian mutation. The control parameters, for example, mutation rate and crossover rate, were set to be the same as [5]. For ABC, the limit parameter is set to be  $SN \times D$ , where  $D$  is the dimension of the problem and  $SN$  is the number of employed bees, which is half of ABC population size. The parameter setting of the ARFO algorithm uses our empirical values summarized in Table 2.

*3.3. Results for Benchmark Functions.* In Table 3 reporting the means and variances of the five algorithms on the ten 30-dimensional test functions for 30 runs is illustrated where best results among involved algorithms are shown in bold.

TABLE 1: Parameters of the test functions.

$f$	Dimensions	Initial range	$x^*$	$f(x^*)$
$f_1$	20	$[-100, 100]^D$	$[0, 0, \dots, 0]$	0
$f_2$	20	$[-30, 30]^D$	$[1, 1, \dots, 1]$	0
$f_3$	20	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]$	0
$f_4$	20	$[-500, 500]^D$	$[420.9867, \dots, 420.9867]$	0
$f_5$	20	$[-600, 600]^D$	$[0, 0, \dots, 0]$	0
$f_6$	20	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
$f_7$	20	$[-100, 100]^D$	$[0, 0, \dots, 0]$	390
$f_8$	20 100	$[-100, 100]^D$	$[0, 0, \dots, 0]$	-450
$f_9$	20	No bounds	$[0, 0, \dots, 0]$	-180
$f_{10}$	20	$[-5, 5]^D$	$[0, 0, \dots, 0]$	-330

TABLE 2: Parameters of ARFO algorithm for simulation and optimization progress.

Simulation setting	Values	Optimization setting	Values
The number of initial roots	1	The number of initial population	20
The maximum number of root tip	200	The maximum number of population	100
$Cr$	0.1	$Cr$	0.1
BranchG	100	BranchG	100
$Nmority$	-1	$Nmority$	-1
$S_{max}$	4	$S_{max}$	4
$S_{min}$	1	$S_{min}$	1
$\alpha_{max}$	$\pi/4$	$\alpha_{max}$	$\pi/4$
$\beta_{max}$	$\pi$	$\beta_{max}$	$\pi$

Figures 1 and 2 demonstrate the performance characteristics in terms of convergence rate regarding the best fitness value of the median run of each algorithm on basic benchmarks and CEC 2005 benchmarks, respectively.

On the two unimodal functions, namely, Sphere and Rosenbrock benchmarks, both ARFO and ABC demonstrated competent performance. It is worthy noted ABC algorithm can consistently obtain the global minimum every time on Sphere. However, for the complex unimodal problem-Rosenbrock, ARFO markedly performed more powerfully than other cases.

In terms of complex multimodal benchmarks including variable-separable Rastrigin, variable-separable Schwefel, and nonseparable Griewank, as shown in Figures 1(c)–1(e), the convergence profiles of ARFO and ABC on these functions were significantly better than the other three algorithms. IN particular, on Schwefel and Griewank, the search performance order is ARFO > ABC > PSO > GA > DE. But ABC showed better performance than ARFO on Rastrigin.

Seven shifted and rotated functions  $f_6 \sim f_{10}$  are considered more difficult to be optimized. From Table 3, ARFO obtained best results on  $f_7$ ,  $f_9$ , and  $f_{10}$  and the search performance order is ARFO > ABC > PSO > GA > DE.

PSO also achieved better results on  $f_8$ . From the graph in terms of convergence rate in Figure 2, it is clearly visible that our proposed ARFO and ABC algorithms achieved significantly better performance on most CEC 2005 benchmarks than PSO, DE, and GA. Meantime ARFO performed a little better than ABC except  $f_6$  when other algorithms miss the global optimum basin. The shifted rotated Griewank function is a good example that ARFO successfully avoids falling into local optima and continues to find better results even after the PSO, GA, and DE seem to have stagnated.

In general, ARFO exhibits more significant advantages in terms of accuracy and convergence rate on most of the tested benchmark functions. This can be explained that ARFO can achieve an appropriate balance between exploration and exploitation by incorporating the regrowing operator and the branching operator because the regrowing operation employs a large elongate-length unit to explore the previously unscanned regions while the branching uses a small elongation-length unit to perform fine-tuning exploitation of the global optimum. Furthermore, the branching operator can generate new individuals to maintain its population diversity, which is a key contributing factor.

## 4. Multilevel Threshold for Image Segmentation by ARFO

**4.1. Entropy Criterion Based Fitness Measure.** The Otsu multithreshold entropy measure [37] proposed by Otsu has been popularly employed in determining whether the optimal threshold method can provide image segmentation with satisfactory results. Here, it is used as the objective function for the involved algorithms and its process can be described as follows

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

Let

$$N = \sum_{i=0}^{L-1} h(i), \quad P(i) = \frac{h(i)}{N}, \quad \text{for } 0 \leq i \leq L - 1 \quad (27)$$

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

TABLE 3: Comparison of results with 30 dimensions obtained by ARFO, ABC, PSO, GA, and DE.

Function		ARFO	ABC	PSO	GA	DE
$f_1$	Mean	$3.85E - 13$	<b>1.01E - 600</b>	$6.86E - 03$	$5.24E - 01$	$4.44E + 00$
	Std	$5.29E - 13$	<b>4.33E - 600</b>	$5.37E - 03$	$1.86E - 01$	$3.89E + 00$
	Min	$6.70E - 14$	<b>2.31E - 600</b>	$2.05E - 03$	$4.14E - 01$	$1.93E + 00$
	Max	$1.18E - 12$	<b>1.21E - 500</b>	$1.54E - 02$	$6.73E - 01$	$1.09E + 01$
$f_2$	Mean	<b>6.54E - 01</b>	$3.22E + 00$	$5.63E + 01$	$2.16E + 02$	$4.83E + 04$
	Std	<b>1.43E + 00</b>	$3.06E + 00$	$5.54E + 01$	$1.15E + 02$	$5.40E + 04$
	Min	<b>0</b>	$1.56E - 01$	$1.68E + 01$	$7.17E + 01$	$1.34E + 04$
	Max	<b>3.25E + 00</b>	$7.51E + 00$	$1.46E + 02$	$3.91E + 02$	$1.38E + 05$
$f_3$	Mean	<b>1.94E + 00</b>	$4.02E + 01$	$6.15E + 01$	$1.08E + 02$	$2.21E + 02$
	Std	<b>1.10E + 00</b>	$1.06E + 01$	$2.72E + 01$	$3.09E + 02$	$9.41E + 01$
	Min	<b>8.19E - 01</b>	$2.35E + 01$	$4.44E + 01$	$5.89E + 01$	$1.09E + 02$
	Max	<b>3.25E + 00</b>	$5.04E + 01$	$1.10E + 02$	$1.34E + 02$	$3.63E + 02$
$f_4$	Mean	<b>6.88E - 04</b>	$2.32E + 03$	$3.62E + 03$	$2.43E + 02$	$2.43E + 03$
	Std	<b>5.74E - 04</b>	$2.24E + 02$	$1.82E + 02$	$1.78E + 02$	$4.91E + 02$
	Min	<b>2.15E - 04</b>	$2.22E + 03$	$3.40E + 03$	$1.22E + 02$	$1.74E + 03$
	Max	<b>1.38E - 03</b>	$2.48E + 03$	$3.92E + 03$	$3.51E + 02$	$2.95E + 03$
$f_5$	Mean	<b>4.51E - 03</b>	$6.81E - 02$	$7.54E - 01$	$3.23E + 00$	$2.95E + 01$
	Std	<b>4.42E - 03</b>	$5.96E - 02$	$2.33E - 01$	$6.27E - 02$	$2.19E + 01$
	Min	<b>0</b>	$1.20E - 02$	$5.27E - 01$	$1.88E + 00$	$8.04E + 00$
	Max	<b>1.04E - 02</b>	$1.72E - 01$	$9.86E - 01$	$3.76E + 00$	$4.89E + 01$
$f_6$	Mean	$6.80E - 14$	<b>3.29E - 14</b>	$5.12E + 02$	$7.72E + 01$	$5.25E + 03$
	Std	<b>2.77E - 14</b>	$2.84E - 14$	$6.36E + 02$	$3.21E + 01$	$3.24E + 03$
	Min	$4.85E - 14$	<b>1.49E - 14</b>	$1.95E + 02$	$6.40E + 01$	$8.39E + 02$
	Max	$1.01E - 13$	<b>5.10E - 14</b>	$1.27E + 03$	$1.20E + 02$	$1.23E + 04$
$f_7$	Mean	<b>1.32E + 00</b>	$6.88E + 00$	$5.71E + 06$	$1.99E + 03$	$2.30E + 09$
	Std	<b>1.83E + 00</b>	$3.47E + 00$	$1.88E + 07$	$5.89E + 03$	$2.50E + 09$
	Min	<b>4.85E - 04</b>	$3.93E + 00$	$3.82E + 02$	$3.15E + 03$	$2.60E + 08$
	Max	<b>3.38E + 00</b>	$1.21E + 01$	$3.13E + 07$	$3.05E + 03$	$3.60E + 09$
$f_8$	Mean	$1.76E + 02$	$7.36E + 02$	<b>1.58E + 02</b>	$1.71E + 04$	$1.20E + 04$
	Std	$3.31E + 02$	$4.74E + 02$	<b>4.54E + 01</b>	$3.96E + 03$	$3.61E + 04$
	Min	<b>1.76E + 01</b>	$5.29E + 02$	$1.39E + 02$	$1.11E + 04$	$8.71E + 03$
	Max	$3.39E + 02$	$1.03E + 03$	<b>1.84E + 02</b>	$1.69E + 04$	$1.46E + 04$
$f_9$	Mean	<b>1.44E + 03</b>	$1.70E + 03$	$4.26E + 03$	$2.06E + 03$	$1.75E + 03$
	Std	$6.29E - 01$	<b>6.07E - 13</b>	$4.25E + 02$	$3.14E + 02$	$4.97E + 01$
	Min	<b>1.41E + 03</b>	$1.64E + 03$	$4.01E + 03$	$1.97E + 03$	$1.64E + 03$
	Max	<b>1.48E + 03</b>	$1.77E + 03$	$4.52E + 03$	$2.16E + 03$	$1.81E + 03$
$f_{10}$	Mean	<b>6.12E + 00</b>	$4.79E + 01$	$2.83E + 02$	$5.41E + 01$	$1.64E + 02$
	Std	<b>5.81E - 01</b>	$1.44E + 01$	$5.34E + 01$	$4.70E + 00$	$2.92E + 01$
	Min	<b>3.80E + 00</b>	$3.15E + 01$	$7.08E + 01$	$5.09E + 01$	$1.48E + 02$
	Max	<b>7.14E + 00</b>	$5.81E + 01$	$3.76E + 02$	$6.17E + 01$	$1.79E + 02$

TABLE 4: Objective values and thresholds by the Otsu method.

Image	$M - 1 = 2$		$M - 1 = 3$		$M - 1 = 4$	
	Objective values	Optimal thresholds	Objective values	Optimal thresholds	Objective values	Optimal thresholds
Avion	$4.11E + 04$	114, 175	$4.35E + 04$	95, 151, 192	$3.91E + 04$	85, 131, 172, 208
House	$3.04E + 04$	109, 175	$3.15E + 04$	87, 136, 183	$2.69E + 04$	76, 121, 155, 191
Lena	$1.77E + 04$	95, 156	$2.11E + 04$	84, 123, 175	$1.96E + 04$	76, 121, 155, 191
Hunter	$1.12E + 04$	53, 118	$1.13E + 04$	37, 88, 144	$1.07E + 04$	33, 76, 121, 156
Mean CPU time	$3.99E + 00$		$8.69E + 01$		$2.42E + 03$	

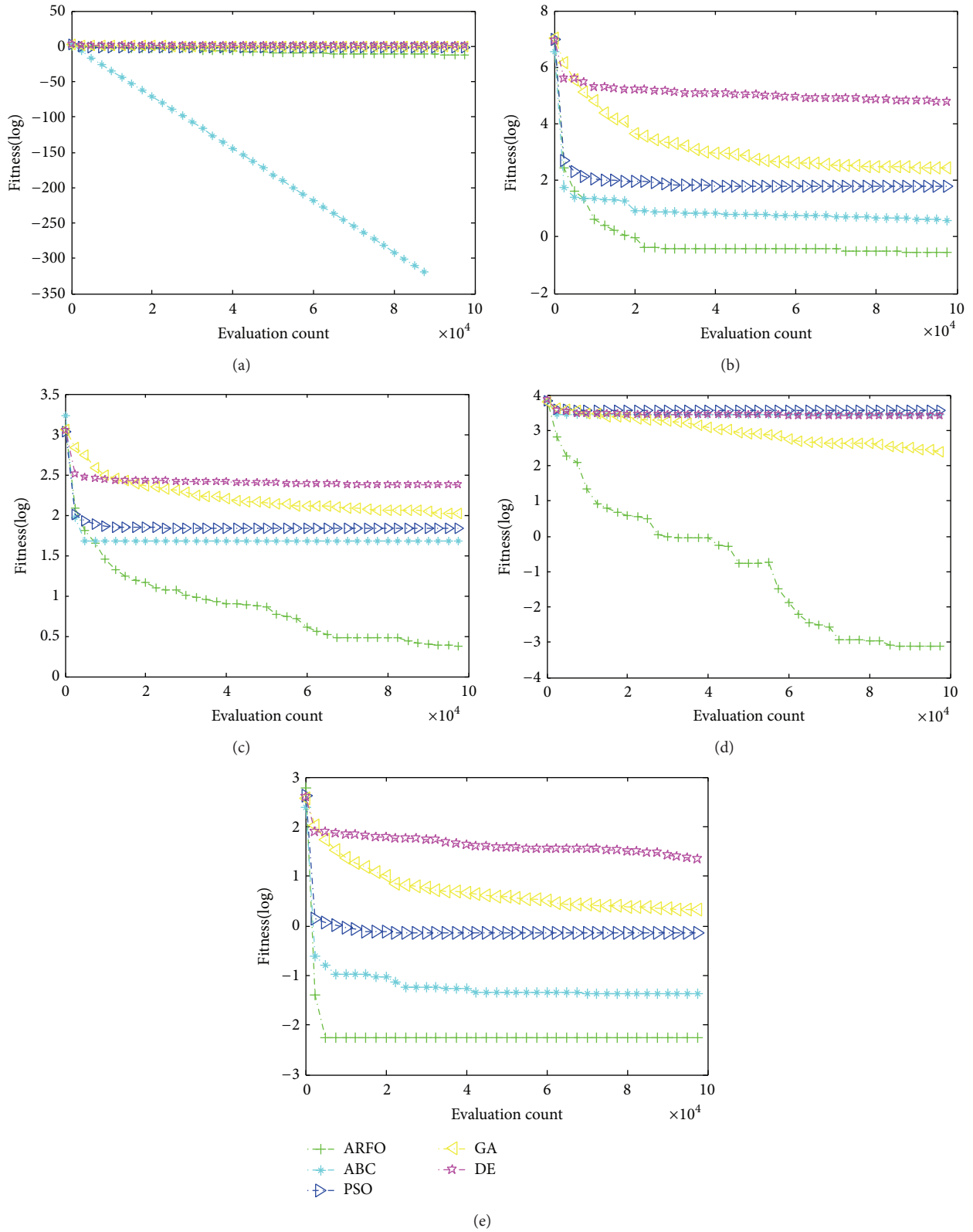


FIGURE 1: Convergence results of ARFO, ABC, PSO, GA, and DE. (a)–(e) correspond to 30-dimensional  $f_1$ – $f_5$ , respectively.



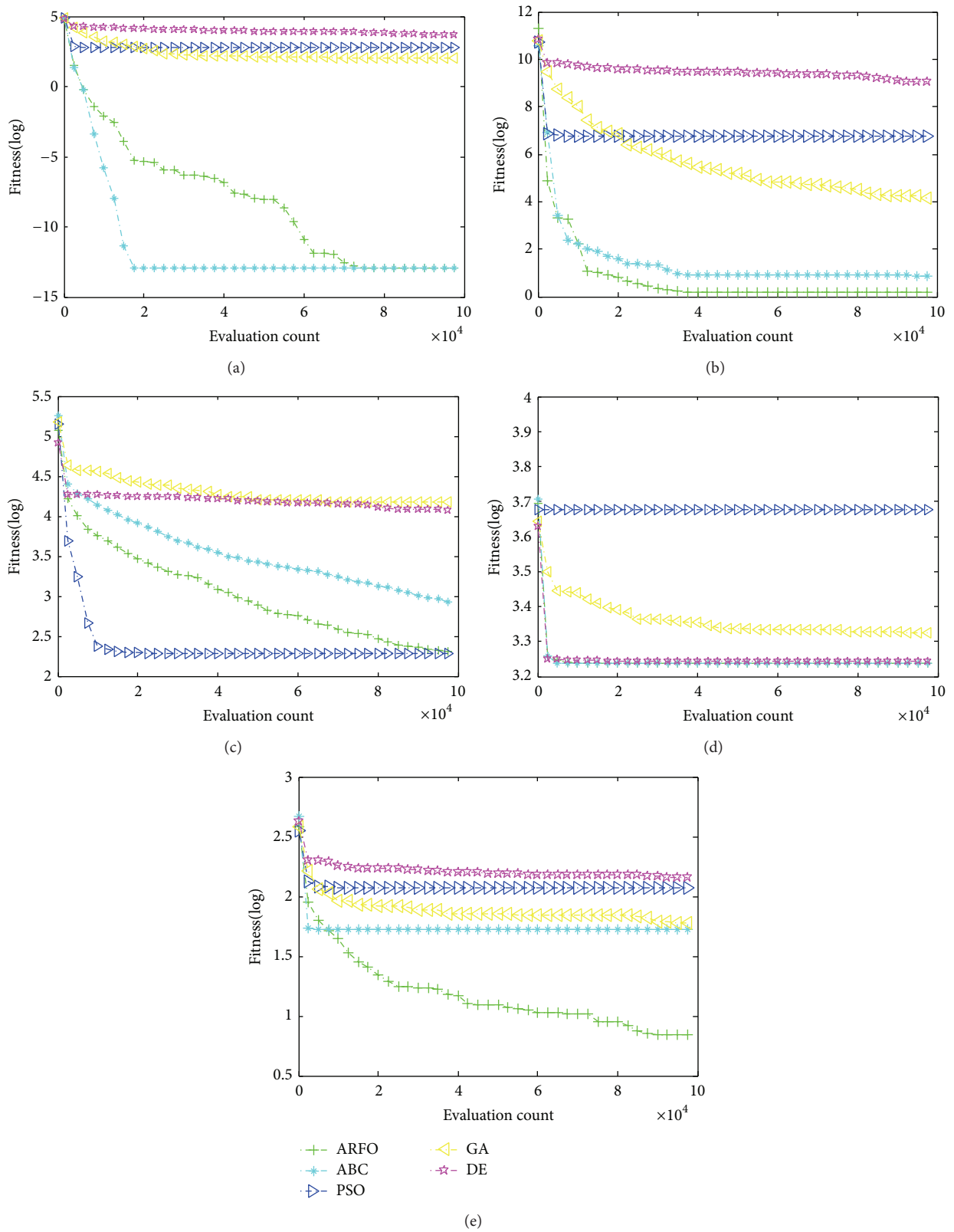
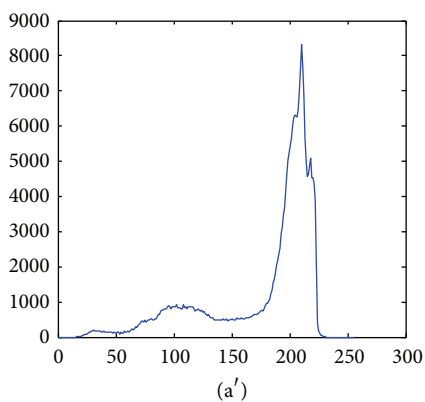


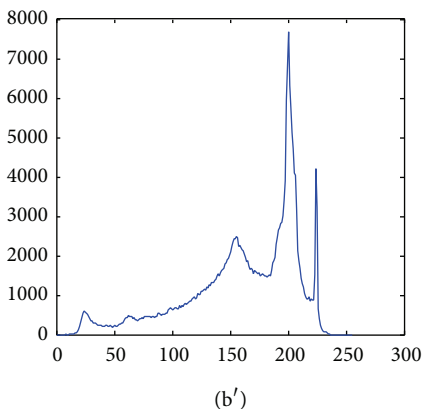
FIGURE 2: Convergence results of ARFO, ABC, PSO, GA and DE. (a)–(e) correspond to 30-dimensional  $f_6$ – $f_{10}$ , respectively.



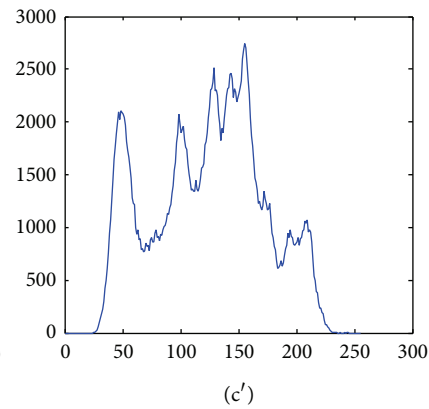
(a) Avion



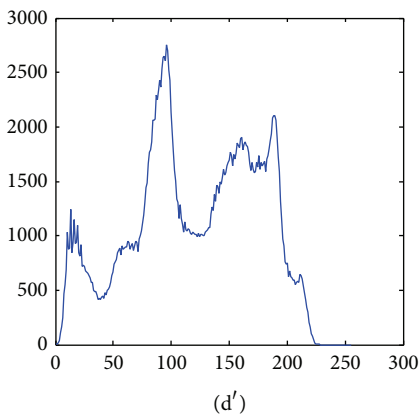
(b) House



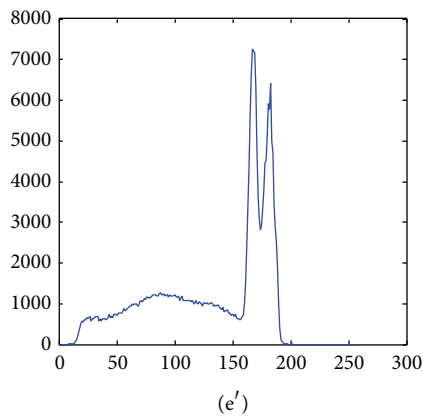
(c) Lena



(d) Peppers



(e) Safari04



(f) Hunter

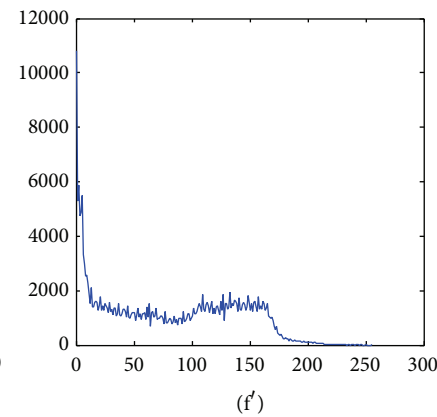
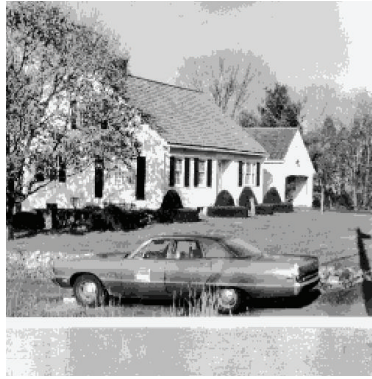


FIGURE 3: Test images and their histograms.





(a) Avion ARFO



(b) House ARFO



(c) Lena ARFO



(a') Avion ABC



(b') House ABC



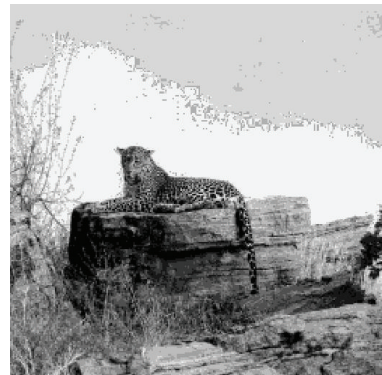
(c') Lena ABC



(a) Hunter ARFO



(b) Peppers ARFO



(c) Safari04 ARFO



(a') Hunter ABC



(b') Peppers ABC



(c') Safari04 ABC

FIGURE 4: The comparison of the thresholded images by ARFO-based and ABC-based algorithms ( $M - 1 = 9$ ).

TABLE 5: The mean CPU time of the compared population-based methods on Otsu algorithm.

Dim	Algorithm				
	ARFO	ABC	PSO	DE	GA
2 Dim	2.54E - 01	3.01E - 01	1.04E + 00	9.55E - 01	3.01E + 00
3 Dim	2.92E - 01	3.44E - 01	1.10E + 00	9.96E - 01	3.76E + 00
4 Dim	2.86E - 01	3.64E - 01	1.11E + 00	1.05E + 00	5.26E + 00
5 Dim	3.26E - 01	3.76E - 01	1.00E + 00	1.10E + 00	1.41E + 00
7 Dim	3.58E - 01	5.26E - 01	1.21E + 00	1.19E + 00	2.04E + 00
9 Dim	5.18E - 01	6.46E - 01	1.23E + 00	1.11E + 00	2.81E + 00

where

$$w_0 = \sum_{i=0}^{t-1} P_i; \quad u_0 = \sum_{i=0}^{t-1} \frac{i \times P_i}{w_0} \quad (28)$$

$$w_1 = \sum_{i=t}^{L-1} P_i; \quad u_1 = \sum_{i=t}^{L-1} \frac{i \times P_i}{w_1} \quad (29)$$

and the optimal threshold is the gray level that maximizes (27). Then (28) can also be written as

$$f(t) = \delta^2 - w_0 \times \delta_0^2 - w_1 \times \delta_1^2, \quad (30)$$

where  $w_0, w_1, u_0,$  and  $u_1$  are the same as given in (9) and (10), and

$$\begin{aligned} \delta_0 &= \sum_{i=0}^{t-1} \frac{(i - u_0)^2 \times P_i}{w_0}, \\ \delta_1 &= \sum_{i=t}^{L-1} \frac{(i - u_1)^2 \times P_i}{w_1}, \\ \delta_0 &= \sum_{i=0}^{L-1} \frac{(i - u)^2 \times P_i}{w}, \\ w &= \sum_{i=0}^{L-1} P_i, \quad u = \sum_{i=0}^{L-1} \frac{i \times P_i}{w}. \end{aligned} \quad (31)$$

Expanding this logic to multilevel threshold,

$$f_{12}(t) = w_0 \times \delta_0^2 + w_1 \times \delta_1^2 + w_2 \times \delta_2^2 + \dots + w_1 \times \delta_N^2, \quad (32)$$

where  $N$  is the number of thresholds.

Equation (32) is considered as the objective function for the proposed ARFO based procedure which is to be optimized (minimized). A close look into this equation will show it is very similar to the expression for uniformity measure.

**4.2. Experiment Setup.** As illustrated above, the performance evaluation based on entropy criterion is one of key indicators in image segmentation scenarios, our experimental studies are conducted on a suit of widely used image datasets [38–43], which contain *avion.ppm*, *house.ppm*, *lena.ppm*, *peppers.ppm*, *safari04.ppm*, and *hunter.pgm* (available at <http://decsai.ugr.es/cvg/dbimagenes/>). The size of each

involved image is  $512 \times 512$ , with a unique grey-level histogram. The intrinsic multimodality of these tested images' histograms will result in a more difficult optimization process. The segmentation results of the proposed algorithm and other algorithms are evaluated by the traditional Otsu method. The parameters of these algorithms (namely ARFO, ABC, PSO, GA, and DE) can be referred in Section 3.2, which have been proven as the optimal setting.

The numbers of thresholds  $M - 1$  investigated in the experiments were 2, 3, 4, 5, 7, and 9. The population size is 20. The maximum number of FEs is 2000, which has sufficient time to search on high dimension problems. Since the population-based algorithms have the randomized characteristics, all the experiments were repeated 30 times for each image for each  $M - 1$  value. In all experiments, to ensure that the initial values of each random algorithm are equal, we used the MATLAB command  $\text{rand}(X, Y) * (\max L - \min L) + \min L$ . The  $\max L$  and  $\min L$  are the maximum and minimum grays of the tested images. Figure 3 presents four original images and their histograms.

**4.3. Experimental Results of Multilevel Threshold.** It is worthy noted the involved population-based algorithms for image segmentation, including ARFO, may only reduce the computation consumption but fail to significantly improve the accuracy of the optimal result. Therefore, we strived to utilize the proposed algorithm to obtain multiple thresholds with larger fitness values and fast computation ability. Due to the exhaustive search characteristic of classical Otsu method, its result can be regarded as the basis for comparison with other EA-based algorithms. We utilized (32) as the fitness function to evaluate involved algorithms.

**Case 1** (multilevel threshold results and efficiency and different methods with  $M - 1 = 2, 3, 4$ ). In tables reporting the fitness, mean computation time, and corresponding optimal thresholds (with  $M - 1 = 2, 3, 4$ ) obtained by Otsu are presented in Table 4. Due to the long consumption of CPU time when  $M - 1 > 4$ , the correlative values are not worthy being listed in our experiment. However, other results attained by the population-based algorithms were compared. It should be noted that in the real-time image proceeding scenarios an appropriate balance of running time and high accuracy is demanded seriously; the termed CPU time of the population-based methods should be analyzed in detail. As shown in Table 5, there are no apparent differences about CPU times among the involved population-based methods,

TABLE 6: Objective value and standard deviation by the compared population-based methods on Otsu algorithm.

Image	$M - 1$	Objective values (standard deviation)				
		ARFO	ABC	PSO	DE	GA
Avion	2	$3.84E + 04$	$3.70E + 04$	$3.83E + 04$	$3.82E + 04$	$3.79E + 04$
		$3.32E - 01$	$8.49E - 12$	$1.58E - 02$	$1.58E - 01$	$3.92E - 01$
	3	$3.84E + 04$	$3.75E + 04$	$3.73E + 04$	$3.84E + 04$	$3.79E + 04$
		$1.82E - 01$	$2.04E - 01$	$1.38E - 01$	$1.58E + 00$	$2.82E - 01$
	4	$3.84E + 04$	$3.79E + 04$	$3.81E + 04$	$3.73E + 04$	$3.77E + 04$
		$5.23E - 01$	$5.39E - 01$	$8.18E - 01$	$3.92E - 01$	$1.63E + 00$
House	2	$3.16E + 04$	$3.04E + 04$	$3.11E + 04$	$3.11E + 04$	$3.11E + 04$
		$3.36E - 02$	$4.80E - 01$	$4.74E - 02$	$3.78E - 03$	$1.05E + 00$
	3	$3.18E + 04$	$3.11E + 04$	$3.07E + 04$	$3.15E + 04$	$3.11E + 04$
		$2.99E - 02$	$3.72E - 02$	$2.04E - 01$	$4.72E - 01$	$7.19E - 01$
	4	$3.23E + 04$	$3.13E + 04$	$3.15E + 04$	$3.15E + 04$	$3.12E + 04$
		$9.14E - 01$	$5.42E - 01$	$7.04E - 01$	$3.61E - 01$	$5.08E + 05$
Lena	2	$1.96E + 04$	$1.92E + 04$	$1.92E + 04$	$1.93E + 04$	$1.91E + 04$
		$9.90E - 03$	$4.17E - 12$	$2.40E - 02$	$3.84E - 12$	$5.18E - 01$
	3	$1.98E + 04$	$1.94E + 04$	$1.96E + 04$	$1.96E + 04$	$1.94E + 04$
		$8.12E - 03$	$5.94E - 02$	$3.60E - 01$	$4.82E - 02$	$1.28E + 00$
	4	$1.98E + 04$	$1.94E + 04$	$1.93E + 04$	$1.95E + 04$	$1.93E + 04$
		$2.34E - 01$	$1.09E + 00$	$4.04E + 00$	$3.92E - 01$	$2.15E + 00$
Peppers	2	$1.93E + 04$	$1.84E + 04$	$1.89E + 04$	$1.87E + 04$	$1.86E + 04$
		$3.78E - 01$	$4.25E - 12$	$3.58E - 01$	$4.71E - 12$	$1.71E - 01$
	3	$1.94E + 04$	$1.87E + 04$	$1.90E + 04$	$1.89E + 04$	$1.89E + 04$
		$9.03E - 01$	$1.50E - 02$	$1.84E - 01$	$5.07E - 01$	$3.82E - 01$
	4	$1.95E + 04$	$1.88E + 04$	$1.93E + 04$	$1.93E + 04$	$1.91E + 04$
		$8.79E - 01$	$1.50E - 01$	$1.51E + 00$	$7.17E - 12$	$4.27E + 00$
Safari04	2	$2.26E + 04$	$2.23E + 04$	$2.23E + 04$	$2.14E + 04$	$2.23E + 04$
		$4.81E - 01$	$9.61E - 01$	$2.50E - 01$	$3.66E - 01$	$3.93E - 01$
	3	$2.28E + 04$	$2.24E + 04$	$2.23E + 04$	$2.15E + 04$	$2.23E + 04$
		$8.34E - 01$	$3.79E - 02$	$1.62E - 01$	$6.49E - 12$	$5.03E - 01$
	4	$2.27E + 04$	$2.23E + 04$	$2.25E + 04$	$2.24E + 04$	$2.23E + 04$
		$5.70E - 01$	$1.11E + 00$	$3.86E - 01$	$1.46E + 00$	$5.03E - 01$
Hunter	2	$1.02E + 04$	$1.01E + 04$	$1.00E + 04$	$1.01E + 04$	$1.01E + 04$
		$1.25E - 01$	$3.56E - 02$	$4.74E - 01$	$2.22E - 04$	$6.26E - 01$
	3	$1.05E + 04$	$1.02E + 04$	$1.02E + 04$	$1.02E + 04$	$1.03E + 04$
		$1.34E + 00$	$7.27E - 01$	$7.07E - 01$	$2.57E + 00$	$1.62E + 00$
	4	$1.05E + 04$	$1.03E + 04$	$1.03E + 04$	$1.04E + 04$	$1.04E + 04$
		$9.16E - 01$	$3.60E + 00$	$1.99E + 00$	$3.63E - 01$	$3.82E + 00$

which are significantly superior in terms of time complexity for high-dimensional image segmentation problems.

From Table 6, it can be observed that the termed fitness values result achieved by the proposed ARFO is generally closed to that by Otsu when  $M - 1 = 2, 3, 4$ , whereas the performance regarding computation time of ARFO is significantly better than its counterpart Otsu. Furthermore, the ARFO-based method also performed most powerful among the population-based methods in most involved datasets. This can be explained by the fact that the ARFO has an appropriate balance between exploration and exploitation. Compared with other population-based algorithms, the ARFO is endowed with branching operations to vary population according to the nutrient concentration change to facilitate better exploitation search in complex higher dimensional search space as well as the branching operation will be used to fine tune the best solutions.

Case 2 (multilevel threshold results and efficiency and different methods with  $M - 1 = 5, 7, 9$ ). In this section, these algorithms are conducted on image segmentation with  $M - 1 = 5, 7, 9$ , which can be considered as higher dimensional problems. In Table 7 reporting the termed experimental results of average fitness value and standard deviation obtained by each population-based algorithm is illustrated, where the larger values and smaller standard deviations indicate the better achievement. From Table 7, it is clearly visible that there are statistically significant differences among experimental results achieved by these population-based approaches, in terms of both accuracy (fitness values) and stability (standard deviation). Depending on the population-varying model and the fast convergence ability, ARFO exhibits more efficient performances than other classical population-based algorithms, especially on high dimensional problems. The results as illustrated in Figure 4 prove that the ARFO-based



TABLE 7: Objective value and standard deviation by the compared population-based methods on Otsu algorithm.

Image	$M - 1$	Objective values (standard deviation)				
		ARFO	ABC	PSO	DE	GA
Avion	5	$3.78E + 04$	$3.73E + 04$	$3.76E + 04$	$3.77E + 04$	$3.73E + 04$
		$3.53E - 01$	$4.63E + 00$	$1.47E + 00$	$2.23E + 00$	$2.41E + 00$
	7	$3.77E + 04$	$3.74E + 04$	$3.76E + 04$	$3.76E + 04$	$3.75E + 04$
		$3.78E - 01$	$4.75E + 00$	$1.47E + 00$	$3.73E + 00$	$2.83E + 00$
	9	$3.78E + 04$	$3.76E + 04$	$3.75E + 04$	$3.77E + 04$	$3.74E + 04$
		$1.45E + 00$	$2.35E + 00$	$2.68E + 00$	$3.36E + 00$	$1.59E - 01$
House	5	$3.14E + 04$	$3.08E + 04$	$3.11E + 04$	$3.12E + 04$	$3.09E + 04$
		$1.46E + 00$	$1.70E + 00$	$1.58E + 00$	$2.22E + 00$	$3.65E + 00$
	7	$3.15E + 04$	$3.12E + 04$	$3.10E + 04$	$3.13E + 04$	$3.10E + 04$
		$1.45E - 01$	$5.97E + 00$	$1.46E + 00$	$2.99E + 00$	$3.76E + 00$
	9	$3.16E + 04$	$3.11E + 04$	$3.09E + 04$	$3.13E + 04$	$3.10E + 04$
		$1.12E + 00$	$5.85E + 00$	$1.25E + 00$	$1.03E + 00$	$1.29E + 00$
Lena	5	$1.97E + 04$	$1.94E + 04$	$1.91E + 04$	$1.95E + 04$	$1.92E + 04$
		$1.58E - 01$	$2.13E + 00$	$1.39E + 00$	$2.78E + 00$	$2.60E + 00$
	7	$1.97E + 04$	$1.94E + 04$	$1.95E + 04$	$1.95E + 04$	$1.93E + 04$
		$1.15E + 00$	$5.74E + 00$	$2.59E + 00$	$1.79E - 01$	$1.48E + 00$
	9	$1.96E + 04$	$1.93E + 04$	$1.94E + 04$	$1.95E + 04$	$1.93E + 04$
		$1.57E + 00$	$6.83E + 00$	$3.76E + 00$	$1.23E + 00$	$3.66E + 00$
Peppers	5	$1.93E + 04$	$1.89E + 04$	$1.89E + 04$	$1.92E + 04$	$1.88E + 04$
		$5.06E - 01$	$1.57E + 00$	$1.79E + 00$	$2.19E + 00$	$4.31E + 00$
	7	$1.93E + 04$	$1.88E + 04$	$1.89E + 04$	$1.92E + 04$	$1.88E + 04$
		$1.39E + 00$	$2.67E + 00$	$4.01E + 00$	$5.07E - 01$	$2.83E + 00$
	9	$1.92E + 04$	$1.89E + 04$	$1.91E + 04$	$1.92E + 04$	$1.90E + 04$
		$1.20E + 00$	$2.69E + 00$	$2.01E + 00$	$5.07E - 01$	$4.39E + 00$
Safari04	5	$2.35E + 04$	$2.20E + 04$	$2.21E + 04$	$2.0445E4$	$2.20E + 04$
		$1.03E - 01$	$6.73E + 00$	$1.46E + 00$	$2.23E + 04$	$2.19E + 00$
	7	$2.34E + 04$	$2.20E + 04$	$2.22E + 04$	$3.54E + 00$	$2.20E + 04$
		$1.48E + 00$	$4.75E + 00$	$1.11E + 00$	$2.23E + 04$	$3.74E + 00$
	9	$2.25E + 04$	$2.20E + 04$	$2.21E + 04$	$2.30E + 00$	$2.20E + 04$
		$9.68E - 01$	$1.43E + 00$	$5.18E - 01$	$3.65E + 00$	$2.34E + 00$
Hunter	5	$1.03E + 04$	$1.02E + 04$	$1.02E + 04$	$1.03E + 04$	$1.02E + 04$
		$7.91E - 01$	$4.32E + 00$	$3.65E + 00$	$4.95E - 01$	$3.61E + 00$
	7	$1.05E + 04$	$1.02E + 04$	$1.02E + 04$	$1.03E + 04$	$1.03E + 04$
		$1.83E + 00$	$3.98E + 00$	$2.35E + 00$	$1.76E + 00$	$2.66E + 00$
	9	$1.04E + 04$	$1.03E + 04$	$1.03E + 04$	$1.03E + 04$	$1.03E + 04$
		$1.37E + 00$	$5.85E + 00$	$2.45E + 00$	$4.94E - 01$	$3.98E + 00$

algorithm is one of the suitable resolutions for multilevel image segmentation problems.

## 5. Conclusions

This paper proposes a new type of bionic algorithm, namely, artificial root foraging optimizer (ARFO) inspired by the plant root behaviors including the tropism and growth behaviors. Through these ecological behaviors, ARFO model can set up the dynamics mechanism of root growing rapidly towards the global optima, in which the auxin concentration

is utilized to determine how to select new growing points and branching number of roots. Then, we introduced three root growing operators to perform the exploring and exploiting progress. Then, against a suit of benchmark functions, ARFO appeared to perform more powerful than other state-of-the-art algorithms in most tested cases. Finally, the ARFO algorithm is applied for the real-world image segmentation problems. The experimental results of ARFO on each dataset indicate its significant improvement compared to other population-based methods. Furthermore, as an effective image segmentation method, ARFO can also be incorporated

into other popular threshold segmentation methods based on optimizing the fitness function.

### Conflict of Interests

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

### Acknowledgments

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

### References

- [1] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [2] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 213, pp. 267–289, 2010.
- [3] D. Corne, M. Dorigo, and F. Glover, *New Ideas in Optimization*, McGraw-Hill, New York, NY, USA, 1999.
- [4] 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.
- [5] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, Mich, USA, 1975.
- [6] M. Dorigo, G. di Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.
- [7] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [8] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micromachine and Human Science*, pp. 39–43, Nagoya, Japan, October 1995.
- [9] G. G. McNickle, C. C. St. Clair, and J. F. Cahill Jr., "Focusing the metaphor: plant root foraging behaviour," *Trends in Ecology and Evolution*, vol. 24, no. 8, pp. 419–426, 2009.
- [10] S. W. Kembel, H. De Kroon, J. F. Cahill Jr., and L. Mommer, "Improving the scale and precision of hypotheses to explain root foraging ability," *Annals of Botany*, vol. 101, no. 9, pp. 1295–1301, 2008.
- [11] Z. Wang, M. van Kleunen, H. J. During, and M. J. A. Werger, "Root foraging increases performance of the clonal plant *Potentilla reptans* in heterogeneous nutrient environments," *PLoS ONE*, vol. 8, no. 3, Article ID e58602, 2013.
- [12] S. K. Gleeson and J. E. Fry, "Root proliferation and marginal patch value," *Oikos*, vol. 79, no. 2, pp. 387–393, 1997.
- [13] C. K. Kelly, "Resource choice in *Cuscuta europaea*," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 89, no. 24, pp. 12194–12197, 1992.
- [14] H. de Kroon, H. Huber, J. F. Stuefer, and J. M. van Groenendael, "A modular concept of phenotypic plasticity in plants," *New Phytologist*, vol. 166, no. 1, pp. 73–82, 2005.
- [15] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.
- [16] T. Pun, "Entropic thresholding: a new approach," *Computer Vision Graphics and Image Processing*, vol. 16, no. 3, pp. 210–239, 1981.
- [17] 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.
- [18] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," *Computer Vision, Graphics, & Image Processing*, vol. 29, no. 3, pp. 273–285, 1985.
- [19] Y. W. Lim and S. U. Lee, "On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques," *Pattern Recognition*, vol. 23, no. 9, pp. 935–952, 1990.
- [20] D. M. Tsai, "A fast thresholding selection procedure for multimodal and unimodal histograms," *Pattern Recognition Letters*, vol. 16, no. 6, pp. 653–666, 1995.
- [21] P. Y. Yin and L. H. Chen, "New method for multilevel threshold using the symmetry and duality of the histogram," *Journal of Electronics and Imaging*, vol. 2, pp. 337–344, 1993.
- [22] A. D. Brink, "Minimum spatial entropy threshold selection," *IEEE Proceedings—Vision, Image and Signal Processing*, vol. 142, no. 3, pp. 128–132, 1995.
- [23] H. D. Cheng, J. Chen, and J. Li, "Threshold selection based on fuzzy c-partition entropy approach," *Pattern Recognition*, vol. 31, no. 7, pp. 857–870, 1998.
- [24] H. Gao, S. Kwong, J. Yang, and J. Cao, "Particle swarm optimization based on intermediate disturbance strategy algorithm and its application in multi-threshold image segmentation," *Information Sciences*, vol. 250, pp. 82–112, 2013.
- [25] H. Gao, W. Xu, J. Sun, and Y. Tang, "Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 4, pp. 934–946, 2010.
- [26] A. Hodge, "The plastic plant: root responses to heterogeneous supplies of nutrients," *New Phytologist*, vol. 162, no. 1, pp. 9–24, 2004.
- [27] L. Dupuy, P. J. Gregory, and A. G. Bengough, "Root growth models: towards a new generation of continuous approaches," *Journal of Experimental Botany*, vol. 61, no. 8, pp. 2131–2143, 2010.
- [28] D. Leitner and A. Schnepf, "Root growth simulation using L-systems," in *Proceedings of the Conference on Scientific Computing (ALGORITMY '09)*, pp. 313–320, 2009.
- [29] D. Eapen, M. L. Barroso, G. Ponce, M. E. Campos, and G. I. Cassab, "Hydrotropism: root growth responses to water," *Trends in Plant Science*, vol. 10, no. 1, pp. 44–50, 2005.
- [30] J. W. Hart, *Plant Tropism and Growth Movement*, Unwin Hyman, 1990.
- [31] N. Takahashi, N. Goto, K. Okada, and H. Takahashi, "Hydrotropism in abscisic acid, wavy, and gravitropic mutants of *Arabidopsis thaliana*," *Planta*, vol. 216, no. 2, pp. 203–211, 2002.
- [32] E. B. Blancaflor and P. H. Masson, "Plant Gravitropism: unraveling the ups and downs of a complex process," *Plant Physiology*, vol. 133, no. 4, pp. 1677–1690, 2003.

- [33] H. M. Leyser, C. A. Lincoln, C. Timpte, D. Lammer, J. Turner, and M. Estelle, "Arabidopsis auxin-resistance gene AXR1 encodes a protein related to ubiquitin-activating enzyme E1," *Nature*, vol. 364, no. 6433, pp. 161–164, 1993.
- [34] S. He, Q. H. Wu, and J. R. Saunders, "Group search optimizer: an optimization algorithm inspired by animal searching behavior," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 973–990, 2009.
- [35] B. Steingrobe, H. Schmid, and N. Claassen, "Root production and root mortality of winter barley and its implication with regard to phosphate acquisition," *Plant and Soil*, vol. 237, no. 2, pp. 239–248, 2001.
- [36] H. Wolpert and W. G. Macready, "No free lunch theorems for search," SFI -TR-95-02-010, Santa Fe Institute, 1995.
- [37] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on System, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [38] W. B. Tao, H. Jin, and L. M. Liu, "Object segmentation using ant colony optimization algorithm and fuzzy entropy," *Pattern Recognition Letters*, vol. 28, no. 7, pp. 788–796, 2007.
- [39] P.-Y. Yin, "Multilevel minimum cross entropy threshold selection based on particle swarm optimization," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 503–513, 2007.
- [40] L. Cao, P. Bao, and Z. Shi, "The strongest schema learning GA and its application to multilevel thresholding," *Image and Vision Computing*, vol. 26, no. 5, pp. 716–724, 2008.
- [41] L. Li, Y. Yang, H. Peng, and X. Wang, "Parameters identification of chaotic systems via chaotic ant swarm," *Chaos, Solitons & Fractals*, vol. 28, no. 5, pp. 1204–1211, 2006.
- [42] H. Peng, L. Li, Y. Yang, and F. Sun, "Conditions of parameter identification from time series," *Physical Review E*, vol. 83, no. 3, Article ID 036202, 2011.
- [43] L. Li, J. Xiao, H. Peng, Y. Yang, and Y. Chen, "Improving synchronous ability between complex networks," *Nonlinear Dynamics*, vol. 69, no. 3, pp. 1105–1110, 2012.



# Hindawi

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

