An Improved Firefly Algorithm with Adaptive Strategies

Xiaohui Yan1.2.*, Yunlong Zhu1, Junwei Wu1.2, and Hanning Chen1

1Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
2Graduate School of the Chinese Academy of Sciences, Beijing, 100039 China

Firefly Algorithm (FA) is a powerful swarm intelligence algorithm inspired by the flash phenomenon of the fireflies. However, it has weaknesses on optimizing high-dimensional problems. This paper presents an improved FA named Adaptive Firefly Algorithm (AFA). In the new algorithm, three strategies are proposed to improve its adaptability and overcome its weaknesses. The algorithm is tested on a set of benchmark functions with different dimensions. Canonical FA and two other well-known heuristic algorithms are also employed for comparison. Experiment and statistical results show that the AFA algorithm offers significant better performance than original FA algorithm. Especially on high-dimensional problems, it is superior to all other compared algorithms.

Keywords: Firefly Algorithm, Adaptive Firefly Algorithm, Adaptive strategies, Particle Swarm Algorithm.

1. INTRODUCTION

Swarm intelligence algorithms are a kind of powerful optimization algorithms inspired by the social behaviors of animal swarms in nature. By cooperation of individuals which have only simple behaviors, complex intelligence could emerge on the level of swarm. Recent years, many swarm intelligence algorithms have been proposed, such as Ant Colony Optimization (ACO),1 Particle Swarm Optimization (PSO),2 Artificial Bee Colony (ABC),3 Bacterial Foraging Optimization (BFO)4 et al. Firefly Algorithm (FA) is a newly proposed swarm intelligence algorithm introduced by professor Yang5 of Cambridge University in 2009. By studying the rhythmic flash phenomenon of the fireflies in the nature, FA algorithm simulates the bioluminescent communication behaviors to optimize problems. It assumes that the fireflies are unisex and move in the space influenced by the flash bright of other fireflies. The algorithm was used for numerical optimization and the results showed that it is superior to some existing meta-heuristic algorithms.6

In Yang’s work, the test functions are all with low dimension and narrow variable ranges. Since the parameters are predetermined, when the dimension and variable ranges increase, the parameters are no longer suitable and the algorithm will perform badly. This will be discussed detail in section 3. In the previous studies, Yang and others researchers proposed a few potential improving strategies.6–7 However, detail modality and experiment have not been given. Some weaknesses still exist. In this paper, some elaborate adaptive strategies are used to improve basic FA algorithm. The novel algorithm is named Adaptive Firefly Algorithm (AFA).

The rest of the paper is organized as follows. In Section 2, we will introduce the original FA algorithm, and its pseudo code is also given. Section 3 will list the shortages when FA algorithm faces the high-dimensional problems. To make up these shortages, several adaptive strategies are introduced and our AFA algorithm is proposed. In Section 4, the AFA algorithm is tested on a set of benchmark functions compared with several other algorithms. Results are presented and statistical methods are used for analyzing. Finally, conclusions are drawn in Section 5.

2. FIREFLY ALGORITHM

There are nearly two thousand firefly species in nature and they inhabit mainly in tropical and temperate regions. Most of the firefly species are capable of producing short and rhythmic flashes when they are flying in the night sky. The bioluminescent behavior is an interesting phenomenon in nature. This feature may be used by the species for courtship rituals, prey attraction, social orientation or even warning the predators.7 Mechanisms of firefly communication via luminescent flashes and has been imitated effectively in some engineering aspects.8 Based on the bioluminescent communication phenomenon of fireflies, Yang proposed the Firefly Algorithm. Three rules are made to idealize the flashing characteristics of fireflies in FA:5

(a) All fireflies are unisex and they will be attracted to other fireflies regardless of their sex.

*Author to whom correspondence should be addressed.
The attractiveness of fireflies to others is proportional to their brightness. So fireflies with less brightness will be attracted by the brighter ones and move towards them.

To certain optimization problem, the brightness of a firefly is determined or affected by the fitness of the solution which the firefly presents in decision space.

In each generation, the fireflies move towards the brighter neighbors according Eq. (1) as followed.

\[ x_i = x_i + \beta_0 e^{-\gamma r_i^2} (x_j - x_i) + \alpha (\text{rand} - 0.5) \]  

(1)

The second term is the brightness attraction. \( \beta_0 \) is maximum attractiveness value, it presents the attractiveness when \( r_{ij} = 0 \). \( \gamma \) is absorption coefficient. \( r_{ij} \) is the distance between firefly \( i \) and firefly \( j \). It makes certain that the attraction of a firefly to another is maximum value when they are at the same place while minimum when the distance is infinite. The third term is the randomization search. \( \alpha \) is the randomization parameter and rand is a random number generated according uniformly distribution in \([0, 1]\). Perturbation is introduced in by this term to enhance the searching ability. The settings of the three parameters are suggested as followed: for most instances, \( \beta_0 = 1, \alpha \in (0, 1), \gamma = 0 \) (1) and \([0.01, 100]\) is recommended.

The pseudo code of FA algorithm is listed in Algorithm 1 (for minimum optimization problems). By the attraction of the brighter ones, all fireflies move towards better area and the optimal point is expected to be found. Numerical tests suggested that FA performs well on multimodal functions and is more powerful than PSO and Genetic Algorithm on some benchmark functions. The pseudo code of FA algorithm is listed in Algorithm 1.

Algorithm 1. Firefly Algorithm

**Inputs:** objective function \( f(x) \); variable boundary; population size \( n \); maximum attractiveness \( \beta_0 \); absorption coefficient \( \gamma \); randomization parameter \( \alpha \)

**Outputs:** best solution

1. **Initialization**
2. \( \text{for } i = 1:n \) 
3. randomly produce \( x_i \) within the variable range
4. \( \text{end if} \)
5. Evaluate the function values of the firefly population
6. \( \text{do while } \) (termination criterion are not met)
7. \( \text{for } i = 1:n \) 
8. \( \text{for } j = 1:n \) 
9. if \( f(x_j) < f(x_i) \)
10. Move firefly \( i \) towards \( j \) according to Eq. 1
11. if \( x_i \) exceeds its boundary 
12. set \( x_i \) to its boundary
13. \( \text{end if} \)
14. \( \text{end if} \)
15. \( \text{end for} \)
16. \( \text{end for} \)
17. Evaluate the new firefly populations
18. Record the best solution achieved so far
19. **end while**

### 3. ADAPTIVE FIREFLY ALGORITHM

Though the FA algorithm is proved to perform better than some heuristic algorithms, it has its limits and weaknesses. Its parameters are all predetermined and they work well on functions with low dimension and narrow variable ranges. However, as the dimension and variable ranges increase, these parameters may not suitable and the algorithm will get worse.

First, consider the attractiveness coefficient, its value \( \beta = \beta_0 e^{-\gamma r_{ij}^2} \). When dealing with the low dimension and small scale problems, the distance between neighbor fireflies are small, the value of \( \beta \) is within a reasonable range. But when dealing with high-dimensional and large scale problems, the distance \( r \) will be too large and the value of \( \beta \) becomes too small. For example, suppose the dimension of an instance is 30, and the average distance of neighbor fireflies on each dimension is 5. The Euclidean distance of two neighbor fireflies is the square root of \((30 \times 5^2)\) and \( r_{ij}^2 = 750 \). Thus the attractiveness coefficient \( \beta = \exp(-750) \) which is nearly zero (with \( \beta_0 = 1 \) and \( \gamma = 1 \)). In fact, the distance mentioned here is just between the adjacent fireflies and the instance listed above is far from the extreme. On problems with high dimension, the points in the space are dispersed, and distance between them are easy enlarged even the variable ranges are not wide. The above \( \beta \) value indicates that no attractions exist even in the adjacent fireflies (all fireflies are short-sighted). Thus the brightness attraction term in Eq. (1) won’t work and all fireflies will move randomly without experience directed. As the brightness attraction is the core motivation in FA algorithm, this will lead to the bad performance on high-dimensional problems. Yang figured that the attractiveness coefficient could be any monotonically decreasing functions such as \( \beta_0 e^{-\gamma r_{ij}^2} (m \geq 1) \). However, the \( r \) value is still too large and \( \beta \) is too small even \( m = 1 \) on high-dimensional problems. We will demonstrate later that \( m \geq 1 \) is not necessary and an adaptive \( m \) parameter would make the algorithm more adaptable.

Second, the randomization term of Eq. (1) is not quite reasonable. Considering that the scales on different dimensions may vary significantly, it is suggested by Yang to use \( S_i \alpha \) instead of \( \alpha \). Where \( \alpha \) is the scaling parameter represents the ranges on all dimensions. This strategy could solve the different scaling problem. Still, other problems exist. An important principle to design a well algorithm is to balance its exploration ability and exploitation ability. The exploitation ability ensures that the algorithm can search the whole space and escape from local optima. The exploitation ability guarantee the algorithm can search carefully and converge to the optimal point. However, the randomization step in Eq. (1) is \([-0.5, 0.5]\) with \( \alpha = 1 \). The perturbation it brought in is too large and weakens the exploitation ability of the algorithm at the late stage of the algorithm, which may prevent the fireflies from converging to the optimal point.

Three adaptive strategies are proposed in AFA to improve the basic FA.

#### 3.1. Adaptive Attraction

As it mentioned above, the attractiveness coefficient \( \beta \) is too small on high-dimensional problems as the distance values are always large. As a result, we need to find a flexible attractiveness coefficient function which could perform well on most problems. \( \beta = \beta_0 e^{-\gamma r_{ij}^2} m \geq 1 \) which is suggested by Yang is a good candidate. With the maximum attractiveness \( \beta_0 = 1 \), the value of \( \beta \) is limited in \([0, 1]\). However, \( m \geq 1 \) is not necessary. When \( m \) is between 0 and 1, the above function is still a monotonically decreasing function, which meets our requirements. More importantly, with \( m \in (0, 1) \), the growth of the function is slowed down gradually so that the value of \( r_{ij}^2 \) is reduced when \( r_{ij} \) is too large.
Accordingly, the $\beta$ value won’t be too small and the brightness attraction term could work well on the high-dimensional problems. For these reasons, we use the above suggested $\beta$ function as the attractiveness coefficient except let $m$ self-adapt according to the problem to be optimized. Based on the weakness and its cause of the algorithm, $m$ should be the function of the dimension and variable ranges, and these are usually known before solving. We suggest using $m = K / \text{Dim} \times \text{Range}$ as our adaptive function. Dim represents the dimension of the problem and Range represents the variable range on one dimension (for problems which scales vary on different dimensions, the largest range is recommended). $K$ is a constant number which $O(1)$ is suggested.

The function is founded based following considerations: (1) The value of $m$ should be inversely proportional to the dimension and the variable ranges; (2) $m$ should be less than 1 on high-dimensional or large range problems. The larger the dimension or range is, the smaller $m$ is. So distances were compressed to maintain $\beta$ in a reasonable range.

3.2. Decreasing Randomization

The randomization term of the FA algorithm will bright out over much perturbation in the late stage. Therefore, decreasing randomization strategy is used in AFA algorithm. Randomization parameter $\alpha$ is no longer a fixed value. Two parameters $\alpha_0$ and $\alpha_{\text{end}}$ are introduced and $\alpha$ decreases from $\alpha_0$ to $\alpha_{\text{end}}$ linearly with iterations. The strategy could keep balance between the exploration and exploitation abilities of the newly algorithm. In the early stage, larger $\alpha$ provides better global searching ability and diversity. And in the late stage, small $\alpha$ value avoids the long-step jumping and offers better convergence. For problems which scales vary on different dimensions, using $\Sigma_j$ multiply by the decreasing $\alpha$ is also suggested.

3.3. Winking Simulation

Add a flash flag for each firefly to identify the firefly is lighting or not at current iteration. It is obvious that a firefly can be seen in the night sky only when it is lighting. A similar assumption is introduced in the FA algorithm: the flight direction of a firefly is affected only by those neighbors who are lighting. As a result, moving condition in AFA algorithm changes as followed: firefly $i$ moves towards firefly $j$ when the fitness of the firefly $j$ is better than $i$ and the $j$th firefly’s flash flag is true (represents the firefly is lighting at current iteration). The flash flags are not sustained. For each firefly, a counter will record the iterations of the firefly remained its current state. The flash flags shift states (true or false) according to a probability $p$ calculated by Eq. 2. count, is the value of the $i$th firefly’s state counter.

\[ p_i = 0.5 + 0.1 \times \text{count}, \]  

Thus is, the larger counts a firefly remains the state, the greater probability the state will shifts. The maximum count a firefly remain its current state is five. After the state changes, the value of corresponding counter will be reset to zero. This bionic strategy makes that each firefly winks in a natural way. In the original FA algorithm, the attraction movements are established when population is produced. However, in the AFA algorithm, randomness is added into the attraction term. A firefly moves towards less-better neighbors when the best neighbor is not lighting now. As a result, this strategy which simulates the winking behaviors can enhance the diversity and avoid premature.

4. EXPERIMENTAL DETAILS

The AFA algorithm was tested on a set of ten benchmark functions. Original FA algorithm, PSO algorithm and a classic evolutionary algorithm- Differential Evolution (DE) algorithm were employed for comparison. Dimensions of these functions were set to be 2, 15 and 30 respectively to test the performances of algorithms on different dimensions. As the results are too much, convergence plots are not given. Instead, statistical techniques were used in this section to analyze the results in a scientific way. There are several different statistical techniques. In this paper, we applied the Iman-Davenport test and the Holm method as a post hoc procedure, which were non-parametric statistical methods and used to analyze the behaviors of evolutionary algorithms in many recent works.

4.1. Benchmark Functions

Ten well-known benchmark functions were used in this experiment, as listed in Table I. The ten benchmark functions are widely adopted by other researchers to test their algorithms in many works. In Yang’s work about FA, the search ranges of some test functions were not given clearly. It is supposed that small ranges were used in these functions according to the results. In this paper, all functions used its standard ranges.

4.2. Parameter Settings for the Involved Algorithms

The population sizes of all algorithms were 40, as it suggested by Yang. In original FA algorithm, all parameters used the default settings: $\beta_0 = 1, \gamma = 1, \alpha = 0.2$. And in AFA, $\beta_0 = 1, \gamma = 1, \alpha_0 = 0.2, \alpha_{\text{end}} = 1.0e - 5, \alpha$ decreased from $\alpha_0$ to $\alpha_{\text{end}}$ linearly with evaluation counts and $\Sigma_j \alpha$ is used. $K = 6$ to keep the $\beta$ function is similar with that of FA on low-dimensional and narrow ranges problems. Standard PSO algorithm was used in this experiment. Inertia weight $\omega$ decreased from 0.9 to 0.7. The learning factors $c_1 = c_2 = 2.0, V_{\text{min}} = 0.1 \times Lb, V_{\text{max}} = 0.1 \times Ub$ where $Lb$ and $Ub$ refer the lower bound and upper bound of the variables. In DE algorithm, $F = 0.85, CR = 1.0$, the differential strategy is “best1jitter”. As the computational complexities may vary from each other for different algorithms, function evaluations (FEs) were used as the measure criterion instead of time. The termination FEs on dimensions of 2, 15 and 30 were 20,000, 60,000 and 100,000 respectively.

4.3. Experiment Results and Statistical Analysis

The mean and standard deviations of function values obtained by the four algorithms with 2, 15 and 30 dimensions are given in Table II, Table III and Table IV. Best values obtained on each function were marked as bold. It is obvious that DE performed best with dimension of 2, AFA algorithm performed best on most functions with dimensions of 15 and 30. FA algorithm performed worst on all three conditions. Though the original FA algorithm achieved acceptable results on most functions with dimension of 2, the other algorithms performed even better. This also confirms the argument that FA is not suitable on functions with high-dimensional and large variable ranges. Results of the four algorithms all change worse with the dimension increased. However, changes of AFA are not too much while changes of FA, DE and PSO are significant, especially the FA and DE.

The Iman-Davenport and Holm tests are used in this section. Details of the two statistical methods are introduced in reference. The average rankings obtained by the four algorithms
with different dimensions are drawn in Figure 1. In the bar diagram, each column with a specific color represents the average ranking obtained by an algorithm with according dimension. The lower the color is, the better its associated algorithm is. It is clear that DE got the best ranking with dimension of 2; AFA got the best ranking with dimensions of 15 and 30.

Table V shows the results of Iman-Davenport statistical test. The values are distributed according to F-distribution with 3 and 27 degrees of freedom. Critical values can be looked up in the F-distribution table and the values listed in Table V are with level of 0.05. With all three kinds of dimensions, the Iman-Davenport values are larger than their critical values, which mean that significant differences exist among the rankings of the algorithms under all three conditions. As a result, Holm test was employed on each condition as a post hoc procedure. With dimension of 2, DE algorithm was chosen as the control algorithm, while with dimensions of 15 and 30, AFA algorithm was the control algorithm. The results of Holm tests with dimensions of 2, 15 and 30 are given in Table VI, Table VII and Table VIII respectively.

Table II. Results of AFA, FA, PSO and DE on benchmark functions with dimension of 2.

<table>
<thead>
<tr>
<th>Function</th>
<th>Formulation</th>
<th>Variable ranges</th>
<th>f(x*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosenbrock</td>
<td>$f_1(x) = \sum_{i=1}^{n} (100(x_i^2 - x_{i+1})^2 + (1 - x_{i+1})^2)$</td>
<td>$[-15, 15]$</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Quadric</td>
<td>$f_3(x) = \sum_{i=1}^{n} (\frac{1}{2} \sum_{j=1}^{n} x_j)^2$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Ackley</td>
<td>$f_4(x) = 20 + e - 20e^{-\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}} - e^{-\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}}$</td>
<td>$[-32.768, 32.768]$</td>
<td>0</td>
</tr>
<tr>
<td>Griewank</td>
<td>$f_5(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$</td>
<td>$[-600, 600]$</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel2.22</td>
<td>$f_6(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>+ \prod_{i=1}^{n}</td>
</tr>
<tr>
<td>Sumsquares</td>
<td>$f_7(x) = \sum_{i=1}^{n} x_i^2$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Powers</td>
<td>$f_8(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>^{i+1}$</td>
</tr>
<tr>
<td>Dixon_Price</td>
<td>$f_9(x) = (x_1 - x_2)^2 + \sum_{i=2}^{n} (2x_i^2 - x_{i+1})^2$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Zakhnov</td>
<td>$f_{10}(x) = \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} 0.5</td>
<td>x_i</td>
<td>^2 + \sum_{i=1}^{n} 0.5x_i$</td>
</tr>
</tbody>
</table>

Table III. Results of AFA, FA, PSO and DE on benchmark functions with dimension of 15.

<table>
<thead>
<tr>
<th>Function</th>
<th>Formulation</th>
<th>Variable ranges</th>
<th>f(x*)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rosenbrock</td>
<td>$f_1(x) = \sum_{i=1}^{n} (100(x_i^2 - x_{i+1})^2 + (1 - x_{i+1})^2)$</td>
<td>$[-15, 15]$</td>
<td>0</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>$f_2(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Quadric</td>
<td>$f_3(x) = \sum_{i=1}^{n} (\frac{1}{2} \sum_{j=1}^{n} x_j)^2$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Ackley</td>
<td>$f_4(x) = 20 + e - 20e^{-\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}} - e^{-\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}}$</td>
<td>$[-32.768, 32.768]$</td>
<td>0</td>
</tr>
<tr>
<td>Griewank</td>
<td>$f_5(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$</td>
<td>$[-600, 600]$</td>
<td>0</td>
</tr>
<tr>
<td>Schwefel2.22</td>
<td>$f_6(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>+ \prod_{i=1}^{n}</td>
</tr>
<tr>
<td>Sumsquares</td>
<td>$f_7(x) = \sum_{i=1}^{n} x_i^2$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Powers</td>
<td>$f_8(x) = \sum_{i=1}^{n}</td>
<td>x_i</td>
<td>^{i+1}$</td>
</tr>
<tr>
<td>Dixon_Price</td>
<td>$f_9(x) = (x_1 - x_2)^2 + \sum_{i=2}^{n} (2x_i^2 - x_{i+1})^2$</td>
<td>$[-10, 10]$</td>
<td>0</td>
</tr>
<tr>
<td>Zakhnov</td>
<td>$f_{10}(x) = \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} 0.5</td>
<td>x_i</td>
<td>^2 + \sum_{i=1}^{n} 0.5x_i$</td>
</tr>
</tbody>
</table>
As shown in Table VI, the p values of FA and AFA are smaller than their α/2 values, which means that equality hypotheses are rejected and significant differences exist between these two algorithms and the control algorithm-DE. The p value of PSO is larger than its α/2 values, so the equality hypothesis can’t be rejected. It denotes that no significant differences exist and it can be regard as equivalent to DE.

With dimension of 15, AFA achieved the best ranking and is the control algorithm. It can be seen from Table VII that the AFA is significant better than FA and DE. Especially the FA, its p value is much smaller than α/2 value. The equality hypothesis of PSO can’t be rejected.

With dimension of 30, AFA is also the best algorithm. It rejected equality hypotheses with FA and DE while didn’t with PSO. This indicates that AFA is significant better than FA and DE. Though the ranking of PSO is worse than FA from Table IV and Figure 1, the difference is not significant by the Holm’s test.

5. CONCLUSIONS

This paper analyzes the weaknesses of Firefly algorithm, especially in the case of high-dimensional and wide range problems. To overcome these weaknesses, an improved firefly algorithm with adaptive strategies is proposed, which named Adaptive Firefly Algorithm (AFA). Three adaptive strategies are introduced in the new algorithm. First, adjust the bright attraction coefficient so that it can self-adapt with problems of different dimensions and variable ranges. Second, randomization parameter decreases linearly with iteration, which could balance the exploration and exploitation abilities to adapt to the needs in different stages of the algorithm. Third, simulate the winking phenomenon to increase diversity and escape from premature.

To verify the optimization ability of AFA algorithm, it is tested on a set of ten benchmark functions with dimensions of 2, 15 and 30. Original FA, PSO and DE algorithms are employed for comparison. The results show that AFA algorithm outperforms FA with all kinds of dimensions. Meanwhile, it performed best among the four algorithms with dimensions of 15 and 30. Iman-Davenport and Holm tests were used for statistic analysis, it also indicate that FA is significant better than FA and DE on these
two kinds of dimension. Overall, the propose AFA algorithm is a competitive algorithm for optimization. It can achieve recommended results and adapt to problems with different dimensions. Especially on high-dimensional problems, it has obvious advantages compared with other heuristic algorithms.

Further research efforts could focus on the following aspects: (1) Test the parameters setting of the AFA in detail. Other forms of bright attraction coefficient could be employed. (2) Apply the algorithm to practical engineering problems.

**Acknowledgments:** This work is supported by the National Natural Science Foundation of China (Grant No. 61174164, 61003208, 61105067). And the authors are very grateful to the anonymous reviewers for their valuable suggestions and comments to improve the quality of this paper.

**References and Notes**


Received: 28 September 2011. Accepted: 15 November 2011.