International Journal on Artificial Intelligence Tools
Vol. 25, No. 6 (2016) 1650032 (30 pages)

DOI: 10.1142/S0218213016500329

AFOA: An Adaptive Fruit Fly Optimization Algorithm with Global Optimizing Ability

Yiwen Zhang*, Guangming Cui†, Erzhou Zhu‡ and Qiang He§
The Key Laboratory of Intelligent Computing & Signal Processing
Ministry of Education, Anhui University, Anhui, China
School of Software and Electrical Engineering
Swinburne University of Technology, Melbourne, Australia
*zhangyiwen@ahu.edu.cn
†cgm133123@163.com
‡ezzhu@ahu.edu.cn
§qhe@swin.edu.au

Received 10 October 2015
Accepted 1 September 2016
Published 28 October 2016

With the development of intelligent computation technology, the intelligent evolution algorithms have been widely applied to solve optimization problem in the real world. As a novel evolution algorithm, fruit fly optimization algorithm (FOA) has the advantages of simple operation and high efficiency. However, FOA also has some disadvantages, such as trapping into local optimal solution easily, failing to traverse the problem domain and limiting the universality. In order to cope with the disadvantages of FOA while retain it merits, this paper proposes AFOA, an adaptive fruit fly optimization algorithm. AFOA adjusts the swarm range parameter V dynamically and adaptively according to the historical memory of each iteration of the swarm, and adopts the more accurate elitist strategy, which is therefore very effective in both accelerating the convergence of the swarm to the global optimal front and maintaining diversity of the solutions. The convergence of the algorithm is firstly analyzed theoretically, and then 14 benchmark functions with different characteristics are executed to compare the performance among AFOA, PSO, FOA, and LGMS-FOA. The experimental results have shown that, AFOA algorithm is a new algorithm with global optimizing capability and high universality.

Keywords: Intelligent evolution algorithm; convergence analysis; fruit fly optimization algorithm (FOA).

1. Introduction

Optimization computation, a long-standing topic based on mathematics, a technique aims at finding a set of parameters to satisfy certain optimal measurements satisfied or making certain performance indexes of the system achieve the optimal solution in the condition of satisfying the specific constraints. Global optimization refers to the procedure of finding the best possible approximate solutions to the relevant objective functions. Nevertheless, with the advent of the characteristics of the optimization problem such as high dimension, strong constraint and multipolarity, it is difficult or even impractical for traditional optimization techniques (such as integer programming,[1,2] dynamic programming,[3] linear programming,[4] nomography[5,6]) to be applied to this new scenario. In recent years, with the rapid development of intelligent optimization algorithm, more and more evolution algorithms such as the genetic algorithm (GA),[7–9] particle swarm optimization (PSO),[10–12] ant colony algorithm (ACO),[13,14] artificial bee colony (ABC)[15–17] and fruit fly optimization algorithm (FOA)[18–20] have been proposed and applied to global optimization computation, which brings a thriving vitality to the optimization computation. However, the common disadvantages of complex computational process and too many parameters make these algorithms hard to be deployed in real applications.[18]

Fruit fly optimization algorithm (FOA) is a novel swarm intelligence evolution algorithm refined from the peculiarity of fruit fly foraging. The algorithm utilizes the unique osphresis and vision of fruit fly as ascendency to optimize, thereby realizing the search of optimal value. This new optimization algorithm has the advantages of being easy to understand and to be written into program code which is not too long compared with other algorithms.[31–33] Therefore, FOA has higher optimization efficiency than other classical PSO, GA for the same number of iterations. However, the algorithm also has a number of defects, such as trapping into local optimal solution easily, failing to traverse the problem domain and limiting the universality.

In order to cope with the disadvantages of FOA while retain it merits, this paper proposes AFOA, an adaptive fruit fly optimization algorithm. The 14 benchmark functions is applied to make more comprehensive and in-depth analysis on the performance of AFOA, according to different characteristic test functions a great deal of contrast experiments are carried out between AFOA and FOA, PSO, LGMS-FOA.[36] Simulation results and comparisons show that AFOA has more stable global search capacity and stronger universality. In particular, our main contributions can be summarized as follows.

(1) An improved quick and effective optimization algorithm, AFOA, is proposed, which adopts dynamic step, historical memory and elitist strategy. It is therefore very effective in both accelerating the convergence of the swarm to the global optimal position and maintaining diversity of the solutions. Furthermore, the new motion equation and smell value of fruit fly swarm are presented.
(2) The convergence condition of fruit fly swarm in AFOA is analyzed. It postulates that random value is constant, which provides theoretical and applied foundation for further improving the global optimization performance of AFOA.

(3) A comprehensive comparison of state-of-the-art evolution optimization algorithms is presented. These benchmark algorithms are often used to evaluate the performance of various new global optimization algorithms.
(4) A host of experimental results and comparisons based on the background of five types of different benchmark functions have been conducted to verify the performance of AFOA by comparing it with several state-of-the-art benchmark algorithms. The experimental results have demonstrated that AFOA algorithm is efficient and has stable global optimizing capability and high universality features.

The rest of this paper is organized as follows. Some related work is discussed in section 2. Section 3 introduces the classic fruit fly optimization algorithm. Section 4 proposes AFOA and analyzes the convergence. Section 5 compares our algorithm with other algorithms. Section 6 conducts a large number of experiments to compare the performance of AFOA with others. Finally, section 7 briefly conclude this paper and outlines our future works.

2. Related Work

Intelligent evolution algorithm is an optimization algorithm to simplify complex issues through simulating the peculiarity of biotic population and utilizing the instinct of organism. However, each intelligent evolution algorithm has more or less limitations as well as defects and needs to be perfected in the ceaseless improvement process, thereby achieving better stability and global search ability. Especially in the multidimensional multi-extremum function optimization, the traditional intelligent algorithm cannot provide satisfactory results. Even so, compared with traditional optimization methods, intelligent evolution algorithm obtains long-term progress and application. Genetic algorithm (GA) is an evolution algorithm originated by Darwinian evolution, which applies genetic operation to intelligent computation and simulates the biotic genetic characteristics.[21–24] But GA has a strong dependence on the initial population and consumes long optimizing time. Particle swarm optimization (PSO) is a novel intelligent evolution algorithm deducing from the simulation of the bird flock foraging and the extraction of its peculiarity.[25–27] PSO also has a strong dependence on the parameter settings, is easy to fall into local optimal and possesses poor global search ability. Ant colony algorithm (ACO) is a colony intelligence evolution algorithm modeled according to the peculiarity of ant's food-seeking, which is inspired by the information exchange between the ant colony.[28–30] But there also exist stagnation phenomenon and slow convergence in ACO. However, the common disadvantages of these stochastic algorithms are complicated computational processes and difficulty of understanding for beginners.

In addition to the intelligent evolution algorithms simulating biological characteristics, there also exist certain evolution algorithms simulating the physical peculiarity of nature. Gravitational Search Algorithm (GSA)[34] proposed by simulating gravitational properties is one of them. The algorithm is an intelligent algorithm through simulating nature's law of universal gravitation and combining vector calculation method. Kinetic Gas Molecule Optimization (KGMO)[35] is proposed by simulating gaseous molecular dynamics. The algorithm is an intelligent algorithm through utilizing the gaseous molecular kinetic energy to vary with the temperature, combining the irregular motion characteristics of gas molecules and using the process of seeking the optimal value in the searching range as model. Nevertheless, such algorithms are so professional, have many complicated problems in calculation process, and thus go against popularization.

3. FOA

FOA algorithm is a new swarm intelligent evolution algorithm abstracted from the peculiarity of fruit fly foraging. In the experiment, fruit fly can determine the food location with its unique osphresis, vision and the odor concentration. Based on this, the optimization process of FOA includes the following parts. Firstly, the odor concentration value of fruit fly individual is utilized to search the extreme value of the swarm. Secondly, the swarm moves to the position of the extreme value and search the new odor concentration, and so forth. Finally, the food position is found, namely the position of optimum solution. The detailed steps of FOA are described as follows.

Step 1. Initializes fruit fly swarm location.

$$\begin{cases} x\_axis = rand(LR) \\ y\_axis = rand(LR) \end{cases} \tag{1}$$

Step 2. Generates the location of fruit fly individual in the swarm.

$$\begin{cases} x_i = x\_axis + rand(V) \\ y_i = y\_axis + rand(V) \end{cases} \tag{2}$$

Step 3. Calculates the distance of each fruit fly individual.

$$Dist_i = \sqrt{x_i^2 + y_i^2} \tag{3}$$

Step 4. Calculates the odor concentration judgement value of each fruit fly individual.

$$S_i = \frac{1}{Dist_i} \tag{4}$$

Step 5. Calculates the odor concentration of each fruit fly individual.

$$Smell_i = Smell\_function(S_i) \tag{5}$$

Step 6. Finds the optimal odor concentration value in the swarm (the maximum value adopted here).

$$[bestSmell \quad bestindex] = max(Smell_i) \tag{6}$$

Step 7. Reserves the optimal extremum which is found and replaces the swarm location.

$$\begin{cases} Smellbest = bestSmell \\ x\_axis = x_{bestindex} \\ y\_axis = y_{bestindex} \end{cases} \tag{7}$$

where LR represents the location parameter of initial swarm and V represents the range parameter generated by the swarm.

4. AFOA — The Proposed Algorithm

AFOA is proposed based on FOA. Combined with the influence of parameter V (Eq. (2)) in FOA upon the overall optimization effect,[36] as for various practical problems the optimal value of V is different. However, it is difficult to find these optimal values, leading FOA has a bit complex operation and limited universality. Based on this, AFOA adjusts value V dynamically and adaptively according to the historical memory of each iteration of the swarm, so as to find the optimal solution better, faster and more stability.

4.1. *AFOA description*

4.1.1. *Algorithm modeling*

(1) Generating equation of swarm range parameter V.

The change of swarm range parameter V utilizes two parameters, the swarm extremum *Pbest* generated by the $(t-1)$th iteration and the global extremum *Gbest* generated by the last $(t-1)$ iterations. The swarm range in the tth iteration is determined according to the two historical memories and the value V in the $(t-1)$th iteration. The calculation equation is as follow.

$$V_j^t = \omega V_j^{t-1} + 2c\delta(P_j^{t-1} - X_j^{t-1}) \tag{8}$$

where the calculation method of P_j^{t-1} is $P_j^{t-1} = Pbest_j^{t-1} + Gbest_j^{t-1}$, i.e.,

$$V_j^t = \omega V_j^{t-1} + 2c\delta(Pbest_j^{t-1} + Gbest_j^{t-1} - X_j^{t-1}) \tag{9}$$

where V_j^t represents swarm range of the jth dimension in the tth iteration, V_j^{t-1} represents swarm range of the jth dimension in the $(t-1)$th iteration, X_j^{t-1} represents swarm location of the jth dimension in the $(t-1)$th iteration, $Pbest_j^{t-1}$ represents swarm extremum of the jth dimension in the $(t-1)$th iteration, $Gbest_j^{t-1}$ represents global extremum of the jth dimension in the first $(t-1)$ iterations, δ denotes random value in $[0,1]$, ω and c denote inertia weight. The calculation equation is shown in Eq. (10).

$$\begin{cases} \omega = \omega_{min} + \dfrac{(\omega_{max} - \omega_{min})(iter - t)}{iter} \\ c = (c_{max} - c_{min})\dfrac{t}{iter} \end{cases} \tag{10}$$

where ω_{max}, ω_{min}, c_{max}, c_{min} are parameters, $iter$ denotes the total number of iterations and t denotes the tth iteration.

(2) Generating equation of the swarm individual.

$$x_{i,j}^{t} = X_{j}^{t-1} + V_{j}^{t} \times \eta\,, \quad (\eta \in [-1, 1]) \tag{11}$$

where X_{j}^{t-1} represents swarm extremum location of the jth dimension in the $(t-1)$th iteration as well as swarm location of the jth dimension in the tth iteration, V_{j}^{t} represents swarm range generated by Eqs. (8) and (9), η denotes random value in $[-1, 1]$, $x_{i,j}^{t}$ represents the location of the jth dimension of the ith fruit fly individual in the tth iteration.

(3) Generating equation of distance and odor concentration judgment value of fruit fly individual.

The calculation equation of fruit fly individual distance is

$$Dist_{i,j}^{t} = x_{i,j}^{t} \tag{12}$$

The calculation equation of odor concentration judgment value is

$$S_{i,j}^{t} = Dist_{i,j}^{t} = x_{i,j}^{t} \tag{13}$$

where $Dist_{i,j}^{t}$ represents distance value of the jth dimension of the ith fruit fly individual in the ith iteration, $S_{i,j}^{t}$ represents odor concentration judgment value of the jth dimension of the ith fruit fly individual in the tth iteration.

4.1.2. *The principle of algorithm improvement*

In the traditional FOA, postulate that the optimization problem contains many maximum points (as is shown in Fig. 1). In the case that the points from the origin to the extreme point A are iterated successively and the swarm range is value V, if the extreme points in this range can be found each time, then it is always

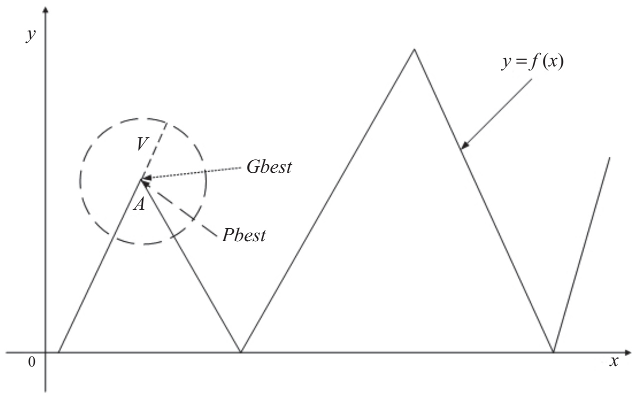

Fig. 1. Optimized function curve.

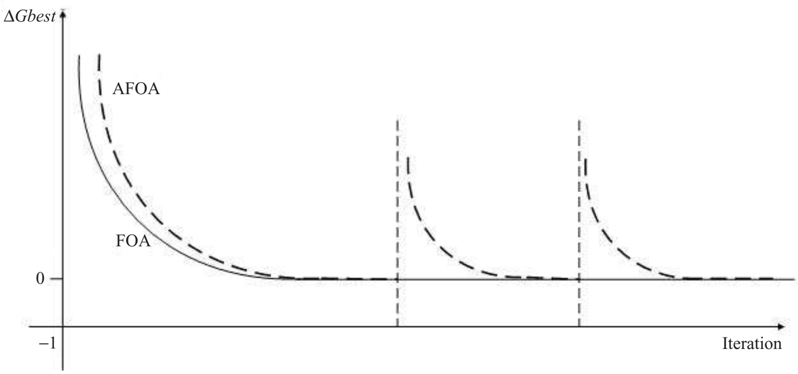

Fig. 2. The analysis for $\Delta Gbest$ between FOA and AFOA.

satisfied that $Pbest = Gbest \wedge \Delta Gbest = Gbest^{t+1} - Gbest^{t}$, i.e., $Pbest - Gbest = 0$. Where $Pbest$ denotes swarm extremum in the current iteration and $Gbest$ denotes global extremum which is found in the swarm. If value V is unaltered or reduced, $Pbest - Gbest = 0 \wedge \Delta Gbest = 0$ is always satisfied, i.e., the optimization converges to the extreme point A. While it can be seen from the figure that the function extreme point is obviously not point A, i.e., the global search ability of the algorithm is weak. However, in the process of value V increasing, there is some probability to jump out of the local optimal position A. The AFOA is just based on this principle. As shown in Eq. (9), while $Pbest = Gbest = X$ (X denotes swarm location and when iterating in point A, $Pbest$ and $Gbest$ are also converted into swarm location.), it can be seen from Eq. (9) that range value V retains at the rate of ω and increases in the number of $2c\delta Pbest$, where parameter 2 is to improve the growth rate and $c\delta$ indicates increase proportion. The change process of FOA and AFOA are simulated with $\Delta Gbest$ as a vertical coordinate, as shown in Fig. 2.

In Fig. 2, solid line indicates the variation trend of FOA and dashed line denotes the variation trend of AFOA. Due to the large difference of initial $\Delta Gbest$ in reality, with the number of the iterations increasing the difference becomes tiny, leading the variation trend cannot be displayed. Therefore, some amplification and simplification are made here, and the variation trend of the difference is simulated. When $\Delta Gbest = 0$, it is indicated that the algorithm finds the extreme point, but it may not be the global extreme point. Through the treatment of Eq. (9), AFOA gets the difference of $\Delta Gbest$ to vibrate again and fluctuate, thereby reaching the next extreme point, over and over again. After all the extreme points are traversed, the optimal extreme point can be found.

4.2. *Algorithm implementation*

In the process of algorithm implementation, due to the presence of domain problems, the location of individuals generated by Eq. (11) may exceed the domain. So,

Algorithm 1 Processing algorithm of the location of individuals $Position(x^t_{i,j})$

Input:
the detailed location of fruit fly individual
Output:
the processed location of fruit fly individual

while $x^t_{i,j} > R_{max}$ **do**
  if $R_{max} > 0$ **then**
    $x^t_{i,j} \leftarrow x^t_{i,j} - R_{max}$
  else if $R_{max} = 0$ **then**
    $x^t_{i,j} \leftarrow 0$
  else
    $x^t_{i,j} \leftarrow x^t_{i,j} + R_{max}$
  end if
end while
while $x^t_{i,j} < R_{min}$ **do**
  if $R_{min} > 0$ **then**
    $x^t_{i,j} \leftarrow x^t_{i,j} + R_{min}$
  else if $R_{min} = 0$ **then**
    $x^t_{i,j} \leftarrow 0$
  else
    $x^t_{i,j} \leftarrow x^t_{i,j} + R_{min}$
  end if
end while

Algorithm 2 Processing algorithm of swarm range V $CalculateV(V^t_j)$

Input:
the swarm range V
Output:
the processed swarm range V

if $V^t_j > 0.4 * R_{max}$ **then**
  $V^t_j \leftarrow 0.4 * R_{max}$
end if
if $V^t_j < 0.4 * R_{min}$ **then**
  $V^t_j \leftarrow 0.4 * R_{min}$
end if

it is needed to process the location of individuals further. The processing algorithm of the location of individuals is described as Algorithm 1.

Where R_{min}, R_{max} denote upper and lower bounds of the domain respectively. In Eq. (9) the swarm range value V may be large or small. In order to achieve better optimization effect, the algorithm processing is performed (Algorithm 2) for V.

Algorithm 3 AFOA

Input:
the test function and the domain

Output:
the minimum which is found in the domain

```
Get(R_min, R_max)
init(X)
for each t ≤ iter do
   for each i ≤ popsize do
      for each j ≤ dim do
         x^t_{i,j} ← X^t_j + V^t_j * θ   (θ ∈ [−1, 1])
         Position(x^t_{i,j})
         Calculate(Dist^t_{i,j}, S^t_{i,j})
      end for
      Calculate(Smell^t_i)
      if bestSmell < Smell^t_i then
         bestSmell ← Smell^t_i
         bestindex ← i
      end if
   end for
   for each i ≤ dim do
      Pbest^t_j ← x^t_{bestindex,j}
   end for
   if bestSmell < Smellbest then
      Smellbest ← bestSmell
      for each j ≤ dim do
         Gbest^t_j ← x^t_{bestindex,j}
      end for
   end if
   for each j ≤ dim do
      V^t_j ← ωV^t_j + 2cδ(Pbest^t_j + Gbest^t_j − X^t_j)
      CalculateV(V^t_j)
      X^t_j ← Gbest^t_j
   end for
end for
```

The processing procedure is to determine whether V is greater than 40% of the domain. If larger, the boundary value is adopted.

The implementation process of AFOA is described as Algorithm 3.

Where *iter* denotes the number of iteration, *popsize* denotes the swarm size, *dim* denotes the test dimension, *Smellbest* represents the fitness value of global extremum that is found, $Gbest_j^t$ represents the location of global extremum that is found, *bestSmell* represents the fitness value of local extremum in the current iteration, $Pbest_j^t$ represents the location of local extremum in the current iteration.

4.3. Analysis of convergence

Theorem 4.1. Consider the equation

$$t_{n+1} + pt_n + qt_{n-1} = b + c_n \tag{14}$$

where $\lim_{n\to 0} c_n = 0$, if the two roots, λ_1 and λ_2, of characteristic equation $\lambda^2 + p\lambda + q = 0$ satisfy $|\lambda_1| \in [0,1), |\lambda_2| \in [0,1)$, any solution $\{t_n\}$ of Eq. (14) converges and $\lim_{n\to 0} t_n = \frac{b}{(1-\lambda_1)(1-\lambda_2)}$.

Proof. Let $t_{n+1} - \lambda_1 t_n = s_{n+1}$, then it can be seen from Eq. (14) that $s_{n+1} - \lambda_2 s_n = b + c_n$.

Let $s_{n+1} - \frac{b}{1-\lambda_2} = r_{n+1}$, then it can be seen that $r_{n+1} = \lambda_2^n r_1 + c_n + \lambda_2 c_{n-1} + \cdots + \lambda_2^{n-1} c_1$.

Let $p_n = c_n + \lambda_2 c_{n-1} + \cdots + \lambda_2^{n-1} c_1$ and postulate that $\forall\, \varepsilon \in (0,\infty), \exists\, N, \forall\, n \geq N$, $|c_n| \in [0,\varepsilon)$, then it can be concluded that for $\forall\, m \in N$, (N denotes the set of natural numbers):

$$\begin{aligned}|P_{N+m}| &= |c_{N+m} + \lambda_2 c_{N+m-1} + \cdots + \lambda_2^{N+m-1} c_1| \\ &\leq |c_{N+m} + \lambda_2 c_{N+m-1} + \cdots + \lambda_2^{m-1} c_{N+1}| + |\lambda_2|^m |c_{N+2} + \cdots + \lambda_2^{N-1} c_1| \\ &\leq \varepsilon \frac{1-|\lambda_2|^m}{1-|\lambda_2|} + |\lambda_2|^m |c_{N+2} + \cdots + \lambda_2^{N-1} c_1|\,.\end{aligned}$$

Then it can be seen that $\lim_{n\to\infty} p_{N+m} = 0 \Rightarrow \lim_{n\to\infty} r_n = 0$. Thus, $t_{n+1} - \lambda_1 t_n = \frac{b}{1-\lambda_2} + r_{n+1}$, where, $\lim_{n\to\infty} r_n = 0$.

Repeat the steps above, we can obtain: $\lim_{n\to\infty} t_n = \frac{b}{(1-\lambda_1)(1-\lambda_2)}$. □

4.3.1. *Analysis on the convergence of swarm location*

If $2c\delta$ is simplified to φ_1 and θ is simplified to φ_2 by considering the variation trend of fruit fly swarm, but not considering the variation of $Pbest, Gbest$ and random value θ, then Eqs. (9) and (11) can be simplified as

$$V^t = \omega V^{t-1} + \varphi_1(P + G - X^{t-1}) \tag{15}$$

$$X^t = X^{t-1} + \varphi_2 V^t \tag{16}$$

From Eq. (16) it can be obtained that

$$V^t = \frac{X^t - X^{t-1}}{\varphi_2} \tag{17}$$

Eq. (17) is induced into Eq. (15) and the iterative relation is extended, then it can be seen that

$$\frac{X^t - X^{t-1}}{\varphi_2} = \omega \frac{X^{t-1} - X^{t-2}}{\varphi_2} + \varphi_1(P + G - X^{t-1}) \tag{18}$$

From Eq. (18) we can obtain:

$$X^t - (\omega + 1 - \varphi_1\varphi_2)X^{t-1} + \omega X^{t-2} - \varphi_1\varphi_2(P + G) = 0 \tag{19}$$

Eq. (19) is equivalent to

$$X^{t+2} - (\omega + 1 - \varphi_1\varphi_2)X^{t+1} + \omega X^t = \varphi_1\varphi_2(P + G) \tag{20}$$

Eq. (20) shows that swarm location meets the two order differential equation and $c_n = 0$. Thus the differential equation satisfies Theorem 4.1. Consequently, it can be seen from Eq. (20) that the characteristic equation is

$$\lambda^2 - (\omega + 1 - \varphi_1\varphi_2)\lambda + \omega = 0 \tag{21}$$

Obviously, two roots λ_1, λ_2 of Eq. (21) are

$$\begin{cases} \lambda_1 = \dfrac{\omega + 1 - \varphi_1\varphi_2 + \sqrt{(\omega + 1 - \varphi_1\varphi_2)^2 - 4\omega}}{2} \\ \lambda_2 = \dfrac{\omega + 1 - \varphi_1\varphi_2 - \sqrt{(\omega + 1 - \varphi_1\varphi_2)^2 - 4\omega}}{2} \end{cases} \tag{22}$$

Then the existing conditions of λ_1, λ_2 is

$$\Delta_X = (\omega + 1 - \varphi_1\varphi_2)^2 - 4\omega \geq 0 \tag{23}$$

Thus, when the following relations are satisfied:

$$\begin{cases} \varphi_1\varphi_2 \geq 0 \\ 1 - \omega \geq 0 \\ 2(\omega + 1) \geq \varphi_1\varphi_2 \end{cases} \tag{24}$$

where $|\lambda_1| \in [0, 1), |\lambda_2| \in [0, 1)$ are in existence. From Theorem 4.1 it can be concluded that X^t converges $\lim_{t\to\infty} X^t = \frac{b}{(1-\lambda_1)(1-\lambda_2)} = P + G$.

4.3.2. *Analysis on the convergence of swarm range*

Similarly, from Eq. (15) it can be seen that

$$X^{t-1} = \frac{\omega V^{t-1} + \varphi_1(P + G) - V^t}{\varphi_1} \tag{25}$$

Eq. (25) is induced into Eq. (16), then it can be seen that

$$\frac{\omega V^t + \varphi_1(P + G) - V^{t+1}}{\varphi_1} = \frac{\omega V^{t-1} + \varphi_1(P + G) - V^t}{\varphi_1} + \varphi_2 V^t \tag{26}$$

From Eq. (26) it can be concluded that

$$V^{t+1} - (\omega + 1 - \varphi_1\varphi_2)V^t + \omega V^{t-1} = 0 \tag{27}$$

Eq. (27) is equivalent to

$$V^{t+2} - (\omega + 1 - \varphi_1\varphi_2)V^{t+1} + \omega V^t = 0 \tag{28}$$

Eq. (20) shows that swarm range meets the two order differential equation and $c_n = 0$. Thus, the differential equation satisfies Lemma 1. Consequently, the characteristic equation is

$$\lambda^2 - (\omega + 1 - \varphi_1\varphi_2)\lambda + \omega = 0 \tag{29}$$

Eq. (29) shows that the characteristics equation is equivalent to Eq. (21). Therefore, when Eq. (24) is satisfied, $|\lambda_1| \in [0, 1), |\lambda_2| \in [0, 1)$ are in existence. From Lemma 1 it can be concluded that V^t converges and $\lim_{t\to\infty} V^t = \frac{b}{(1-\lambda_1)(1-\lambda_2)} = 0$.

5. Algorithms Comparison

In order to interpret AFOA better and analyze the essential characteristic of AFOA further, we compare this algorithm with related evolutionary algorithms, including particle swarm optimization (PSO), fruit fly optimization algorithm (FOA) and LGMS-FOA.

5.1. *AFOA versus FOA*

Fruit fly optimization algorithm (FOA) is a novel optimization method inspired by the foraging behavior of fruit fly. The fruit fly itself is superior to other species in sensing and perception, especially in osphresis and vision. Based on it and fruit fly foraging process, fruit fly optimization algorithm is proposed.[24] The implementation of FOA is as follow (Algorithm 4).

Algorithm 4 Implementation process of FOA

```
Step 1. init(X_axis, Y_axis)
Step 2. Get(X, Y)
Step 3. Calculate(Dist, S)
Step 4. Smell_i = fitness_function(S_i)
Step 5. [bestSmell   bestindex] = min(Smell)
Step 6. Set(X_axis, Y_axis)
Step 7. Repeat (Step 2–Step 6)
```

Note: where *fitness_function* denotes fitness function.

AFOA is the improved algorithm based on FOA, whereas AFOA strives to overcome the shortcomings of FOA and makes numerous improvements. The detailed differences are listed as below.

(1) The adopted models differ. FOA is an intelligent algorithm based on fruit fly foraging pattern, while AFOA is an improved intelligent evolutionary algorithm

based on FOA according to the problem of multidimensional multi-extrema function optimization.

(2) The moving equations of fruit fly individual differ. FOA calculates the new location by Eq. (2), while AFOA adopts the more accurate elitist strategy and generates new individual by Eqs. (8)–(10).

(3) The optimizing ways differ. AFOA considers the effects that evolutional contemporary optimal particle and optimal particle generated impose on the whole optimization. Whereas, FOA does not take it into account.

(4) The moving step length V differ. FOA adopts fixed moving step length V and produces new location by using $V * \sigma\,(\sigma \in [-1,1])$. AFOA adopts dynamic adaptation to shift V according to two historical information $Pbest$ as well as $Gbest$, and then produces new position by using the altered $V * \delta\,(\delta \in [-1,1])$.

(5) The definitions of distance differ. FOA adopts the two-dimensional distance equation based on rectangular plane coordinate system, while AFOA utilizes linear processing and generates the judgment value of distance and odor concentration by Eqs. (12) and (13).

5.2. *AFOA versus LGMS-FOA*

LGMS-FOA is also a kind of improved FOA. This algorithm analyzes the defects of classical FOA in nonlinear processing, uses the linear processing method to improve FOA and achieves good experimental results. According to the improved scheme in Ref. 36, the experimental procedures are shown as Algorithm 5.

Algorithm 5 Implementation process of LGMS-FOA

```
Step 1. init(maxgen, popsize, ω0, α, n, gen = 0, x_axis)
Step 2. Set(ω = ω0 * α^gen, x_i = x_axis + ω * rand())
Step 3. Calculate(S_i = x_i)
Step 4. Calculate(Smell_i)
Step 5. Keep-min(Smell_i), Update(x_axis)
Step 6. Repeat (Step 2–Step 5)
```

Note: where $Keep\_min(Smell_i)$ represents obtaining the minimum of swarm odor and reserve it.

Both of these algorithms are improved algorithms based on FOA. Nevertheless, they have different emphases in improvement, different results, different application occasion and different global search capability. The major differences are listed below.

(1) The moving equations of fruit fly individual differ. LGMS-FOA calculates the new location by Algorithm 5, i.e., the producing method is $x_i = x\_axis + \omega_0 * \alpha^{gen} * rand()$. While AFOA adopts the more accurate elitist strategy and generates new individual by Eqs. (8)–(10).

(2) The optimizing ways differ. AFOA considers the effects that evolutional contemporary optimal particle and optimal particle generated impose on the whole optimization. Whereas, LGMS-FOA does not take it into account.
(3) The moving step length V differ. LGMS-FOA adopts variational moving step length V and produces new location by using $\omega_0 * \alpha^{gen} * rand()$. AFOA adopts dynamic adaptation to shift V, considers the impacts of the two historical information $Pbest$ as well as $Gbest$ on the swarm, and then produces new position by using the altered $V * \delta\,(\delta \in [-1, 1])$.

5.3. *AFOA versus PSO*

PSO algorithm proposed by Eberhart and Kennedy in 1995,[37] is a heuristic evolutionary computing method and derives from the simplified simulation of the social group's intelligent behavior. Each particle location represents a solution to optimization problem in PSO. The particle's location is updated according to the two extremums. One is *pbest*, historical optimal solution of the particle itself. The other is *gbest*, global optimal solution which has been found. The location renewal equation of PSO is as follows.[35]

$$v_i^d(t+1) = \omega(t)v_i^d(t) + C_1\, rand_i(t)(pbest_i^d - x_i^d(t)) + C_2\, rand_i(t)(gbest^d - x_i^d(t)) \tag{30}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{31}$$

where $pbest_i = (pbest_i^1, pbest_i^2, \ldots, pbest_i^n)$, $gbest = (gbest^1, gbest^2, \ldots, gbest^n)$ represent historical extremum of individual fly i and historical extremum of fly swarm in the last t evolutionary process, $rand_i(t)$ denotes the random value in the tth evolutionary range of $[0, 1]$ and C_1 as well as C_2 denote two inertia weights.

Both of the two algorithms, belonging to swarm intelligent optimization algorithms, are evolutionary algorithms deduced from nature animal's rule of foraging and have high searching efficiency in the search space. They could obtain the optimal solution by evolving in the next time according to the change of particle location. Nevertheless, the two algorithms have different points in the following aspects.

(1) The proposed backgrounds differ. AFOA is a new global optimization method deduced from foraging behavior of fruit fly. PSO, an optimization algorithm deduced from birds' prey.
(2) The optimizing ways differ. In AFOA, a fruit fly find a better position, then the entire swarm move to the position. While in PSO, each individual changes its location dynamically through each individual optimization and global optimization.
(3) The moving equations differ. AFOA uses swarm location and movement speed to produce individual fruit fly in a certain range, while PSO shifts speed and location by using Eqs. (30) and (31).

(4) The biological characteristics differ. AFOA calculates the smell value by using the distance parameter of fruit fly individual, and then uses the smell value to optimize. While PSO does not take the distance into account and optimizes by using information exchange between the particles to change the location.

6. Experimental Analysis

To make more comprehensive and in-depth analysis on the performance of AFOA, according to different test functions a great deal of contrast experiments are carried out between AFOA and FOA, PSO, LGMS-FOA. This section consists of three parts. The first part detailed lists the information of test functions; the second part is the results of comparison experiments among AFOA, FOA, LGMS-FOA and PSO; and the third part is the comparative analysis and the discussion about the performance of each algorithm.

6.1. *Benchmark function*

To have full verification to the stable global search ability of AFOA, we choose 14 benchmark functions[34–36] and the test functions are shown in Table 1. According to the test functions, the performance of AFOA can be analyzed in the case of multidimensional multi-peak composition.

6.2. *Parameter setting and experimental result*

This experiment verifies the performance of AFOA by many repeated experiments. In order to ensure that each algorithm can find the global optimal solution, the constant results over 50 trials in repeated experiments is used as termination criterion and the average value is used as experimental result. The standard deviation is calculated by each experimental result x_i and the known minimum x_{opt} in Table 1 according to Eq. (32), where n denotes the number of experiments.

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i - x_{opt}^2)}{n}} \tag{32}$$

The experimental environment is windows7 OS, Intel Core i5 dual core 3.2 GHz, 8 G memory, 64 bit operating system. The programming environment is Microsoft visual studio 2010. The experimental parameters in each algorithm are shown in Table 2.

According to the parameters in Table 2 set, combined with the test functions in Table 1 the average results over 50 trials are shown as Table 3.

It is apparent from Table 3 that as a whole, in the aspect of the global search ability the optimization result of AFOA is superior to that of FOA, LGMS-FOA and PSO. Standard deviation of AFOA is a little worse than PSO in certain test functions. However, given that the minimum of some test functions in the domain of Table 1 is not an integer, and given that retaining accuracy and adopting rounding

Table 1. Benchmark functions.

Test Function	Dimension	Domain	Minimum
$F1(X) = \|x\|$	1	$[-10, 10]$	0
$F2(X) = sim(x)$	1	$[-10, 10]$	-1
$F3(X) = x\sin(x)$	1	$[0, 20]$	-17.3076
$F4(X) = \sum_{i=1}^{n} x_i^2$	30	$[-10, 10]^n$	0
$F5(X) = \sum_{i=1}^{n} x_i^2$	50	$[-10, 10]^n$	0
$F6(X) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$	30	$[-10, 10]^n$	0
$F7(X) = \sum_{t=1}^{n} \left(-x\sin(\sqrt{\|x_i\|})\right)$	30	$[-500, 500]^n$	-12569.5
$F8(X) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12, 5.12]^n$	0
$F9(X) = -20\exp\left(-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}\right) - \exp\left(\frac{\sum_{i=1}^{n} \cos 2\pi x_i}{n}\right) + 20 + e$	30	$[-32, 32]^n$	0
$F10(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\frac{x_i}{\sqrt{i}} + 1$	30	$[-600, 600]^n$	0
$F11(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $* [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^2$	3
$F12(X) = \left(x_2 - \frac{5.1}{4 * \pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(2 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$-5 \leq x_1 \leq 10;\ 0 \leq x_2 \leq 15$	0.3979
$F13(X) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2$	2	$[-1, 1]^2$	-2
$F14(X) = \left[\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right] * \left[\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right]$	2	$[-10, 10]^2$	-186.7309

Note: where the minimum value refers to the extremum of the function in the domain and is rounded to 4 decimal places.

Table 2. Parameter settings.

Algorithm	Parameter	Value	Significance
FOA	V	200	The range size of fly swarm
AFOA	ω_{max}	0.95	maximum of inertia weight
	ω_{min}	0.1	minimum of inertia weight
	c_{max}	2	maximum of learning factor
	c_{min}	0.5	minimum of learning factor
LGMS-FOA	n	0.005	search coefficient
	ω_0	1	initial weight
	α	0.95	weight coefficient
PSO	ω_{max}	0.9	maximum of inertia weight
	ω_{min}	0.1	minimum of inertia weight
	c_1	2	local learning factor
	c_2	1.5	global learning factor

Table 3. The experimental results.

	AFOA		FOA		LGMS-FOA		PSO	
Function	Minimum	SD	Minimum	SD	Minimum	SD	Minimum	SD
$F1$	0	0	4.0226e-7	4.0239e-7	8.9328e-226	0	0	0
$F2$	−1	1.2379e-7	−0.0007	0.9993	−1	0	−1	0
$F3$	−17.3076	8.3465e-6	7.2686e-12	17.3076	−17.2008	0.2967	−17.3076	3.2957e-5
$F4$	0	0	0.0049	0.005	4.9407e-324	0	2.7513e-165	0
$F5$	0	0	0.0024	0.0024	1.9763e-323	0	1.3537e-17	9.0714e-17
$F6$	0	0	5.6391e-5	5.6395e-5	0.1088	0.4993	6.0593e-35	3.2370e-34
$F7$	−5373.4159	7220.5062	−0.3608	12569.1392	−6707.8049	5886.7805	−419.73629	12149.7637
$F8$	0	0	0.2521	0.2530	84.0141	86.5788	8.0592	8.5184
$F9$	4.6363e-15	4.833e-15	7.6478e-6	7.6481e-6	0.5325	0.9019	3.9968e-15	3.9968e-15
$F10$	0	0	1.3179e-12	1.3184e-12	0.0109	0.0172	0.0008	0.0029
$F11$	3.0000	0.0001	459.1695	509.0312	3.0000	9.3136e-14	61.3025	68.7103
$F12$	0.3979	0.0001	6.6954	7.1.97	0.4404	0.0686	0.3979	1.2642e-5
$F13$	−2	0	−2	6.7767e-11	−2	0	−2	0
$F14$	−186.7187	0.022	−16.7197	167.0114	−186.7309	8.8307e-6	−186.7309	8.831e-6

off cause optimal value and actual optimal value differ a little, it is acceptable that standard deviation is slightly unsatisfactory. As for the overall experimental result, AFOA has more stable global search capability, meets the requirements of various types of test functions and possesses strong universality.

6.3. *Analysis and discussion on the experimental results*

In accordance with the dimension and the number of peaks, test functions listed in Table 1 can be classified into five categories, including single-dimensional single-peak function, single-dimensional multi-peak function, multidimensional single-peak function, multidimensional multi-peak function and multidimensional composite multi-peak function. In test functions of Table 1, function $F1$ belongs to single-dimensional single-peak function, function $F2$ and $F3$ belong to single-dimensional multi-peak function, function $F4$ to function $F6$ belong to multi-dimensional single-peak function, function $F7 \sim F10$ belong to multidimensional multi-peak function and function $F11 \sim F14$ belong to multidimensional composite function. This section will discuss the global search performance of AFOA compared with FOA, LGMS-FOA and PSO respectively in accordance with the five types of test functions.

6.3.1. *Results analysis on single-dimensional single-peak test function*

Owing to the fact that single-dimensional single-peak function is common and relatively easy to comprehend, there is only function $F1$ belonging to single-dimensional single-peak function in test functions of Table 1. These function extreme points are also minimum points. All the classical evolutionary algorithms can find the extreme points and the only difference is the different accuracy. As for such functions, global search ability is embodied in the precision and searching speed. In 50 times repeated experiments, the global extremum that is found in each iteration is recorded, as shown in Fig. 3.

It can be seen from Table 1 that in the domain the minimum of function $F1$ is 0. It can be seen from Fig. 3 that each of AFOA, PSO, FOA and LGMS-FOA can find a point that is close to the minimum point 0, but the precisions are quite different. The accuracy of FOA is minimal and the precisions of AFOA and PSO are maximal, both of which can accurately find the minimum point 0. The experimental results of different algorithms have large gap. Therefore, if we do not adopt the logarithm (base 10) to process, optimization process cannot be displayed in the image normally. However, the optimization result of AFOA is 0 after processing and the corresponding result does not exist, causing that in Fig. 3 only half the curves of AFOA and PSO can be displayed. It shows that AFOA can find the global minimum point and the image of test functions below also have this phenomenon. So in terms of accuracy, AFOA shows stable global search ability. Considering the progress, in the aspect of searching the speed of stable minimum that is found it can be seen from Fig. 3 that the number of iterations AFOA requires is the

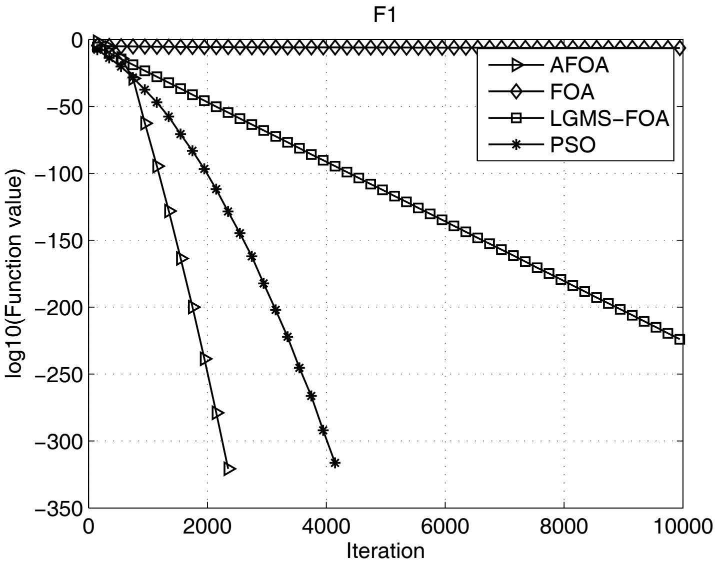

Fig. 3. Comparison of optimization process of F1.

least and the convergence speed of AFOA is the fastest. Consequently, in aspect of single-dimensional single-peak function AFOA has excellent stable global search ability.

6.3.2. *Results analysis on single-dimensional multi-peak test function*

Single-dimensional multi-peak function is also common and relatively easy to understand. There are function $F2$ and $F3$ belonging to this kind of function in test functions of Table 1. These function extreme points are not unique and not necessarily minimum points, where extreme points of function $F2$ is minimum points and not all the extreme points of function $F3$ are minimum points. All the classical evolutionary algorithms can find the extreme points, but extreme point is not necessarily minimum points and the precisions of the results vary greatly. In 50 times repeated experiments, the global extremum that is found in each iteration is recorded, and the experimental results of function $F2$ and $F3$ are shown in Figs. 4 and 5.

Overall, Figs. 4 and 5 show that the optimization precisions of AFOA, PSO and LGMS-FOA significantly outperform that of FOA, especially when the result is negative, classical FOA reveals fatal flaws. Combined with optimization results the optimization effect and stability of AFOA is superior to those of PSO and LGMS-FOA. In terms of convergence speed, although AFOA is slightly worse than LGMS-FOA and PSO in Fig. 5, combined with optimization effect the accuracy of AFOA is higher than those of other algorithms. Consequently, in general AFOA has stable global search ability in aspect of single-dimensional multi-peak function.

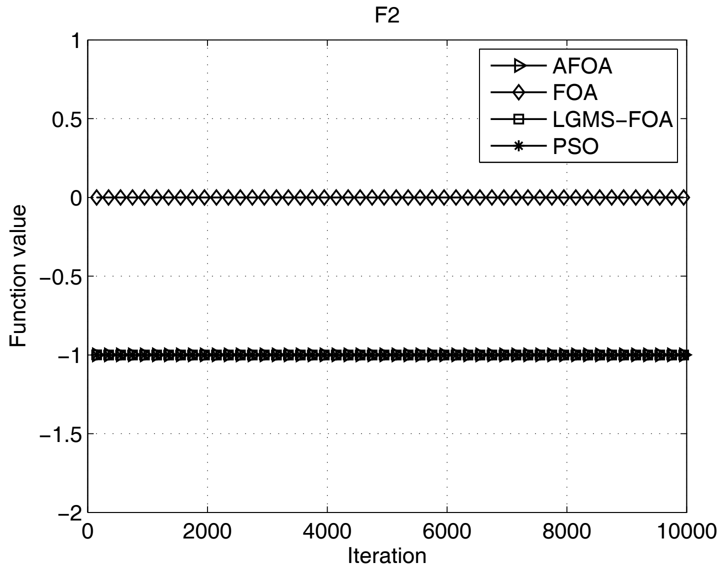

Fig. 4. Comparison of optimization process of F2.

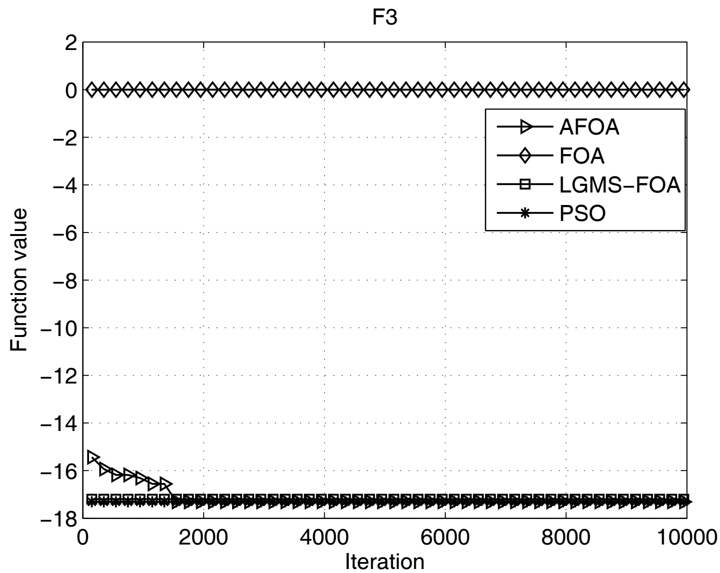

Fig. 5. Comparison of optimization process of F3.

6.3.3. *Results analysis on multidimensional single-peak test function*

Function $F4 \sim F6$ belong to multidimensional single-peak function. These function extreme points are also minimum points. So most of the algorithms can find their minimum and there are just exist differences in precision. Nevertheless, as for global

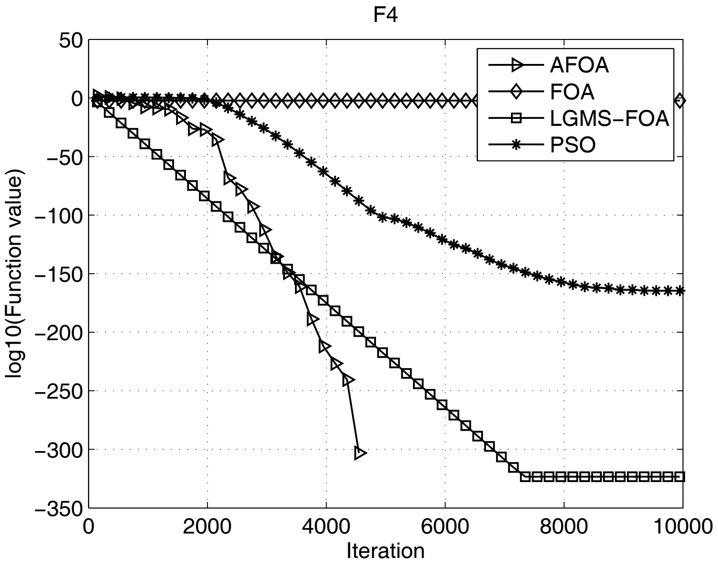

Fig. 6. Comparison of optimization process of F4.

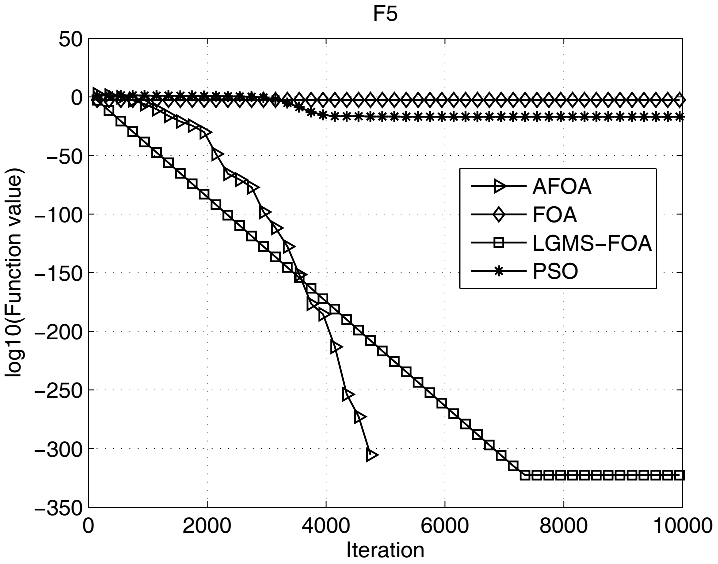

Fig. 7. Comparison of optimization process of F5.

search capability, the accuracy and speed of stable extremum which is found are the most reliable measurement indexes. For function $F4$ to $F6$ in 50 times repeated experiments, the global extremum that is found in each iteration is recorded, as shown in Figs. 6 and 8.

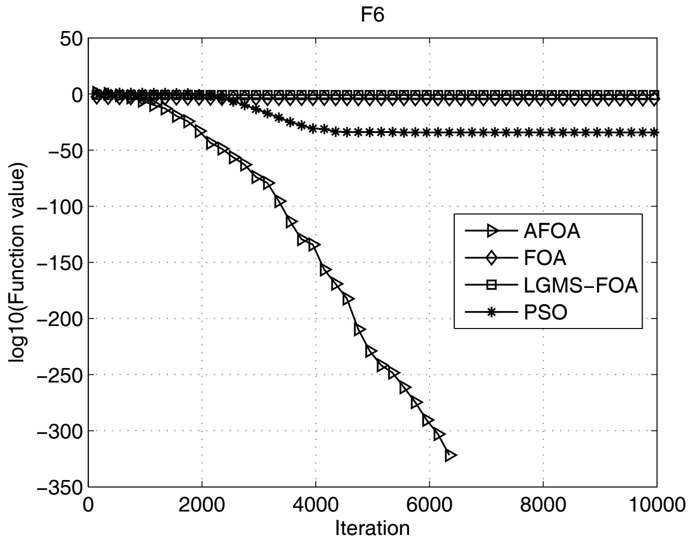

Fig. 8. Comparison of optimization process of F6.

Table 1 shows that all the minimums of test function $F4$ to $F6$ in the domain are 0. Figures 6–8 show that AFOA, FOA, LGMS-FOA and PSO can all find the extreme point. However, in Figs. 6 and 7 the precisions of AFOA and LGMS-FOA are significantly outperform those of FOA and PSO. In Fig. 8 the accuracy of AFOA is significantly higher than those of LGMS-FOA, FOA and PSO. Therefore, in the aspect of multidimensional single-peak function AFOA has good optimization accuracy. In consideration of the high precision it can be seen from Figs. 6–8, AFOA can achieve stable extreme points fastest and has high convergence speed. Consequently, through the analysis on optimization accuracy and convergence speed AFOA has efficient stable global search ability in aspect of multidimensional single-peak function.

6.3.4. *Results analysis on multidimensional multi-peak test function*

The fourth type is multidimensional multi-peak function, which test functions $F7$ to $F10$ belongs to. These functions have the characteristics such as high dimension and multi extremum, and the extreme points are not necessarily minimum points. Such functions have complicated form and is difficult to comprehend. What's worse, classical algorithm is easy to fall into local optimum, thereby having weak global optimization ability. Although finding the minimum, there also exist the problems such as poor precision and slow convergence speed. According to test function $F7$ to $F10$ in 50 times repeated experiments, the global extremum that is found in each iteration is recorded and the experimental results is shown in Figs. 9–12.

Through analyzing the experimental results of four test functions, as a whole AFOA has higher accuracy and faster convergence speed when meeting the

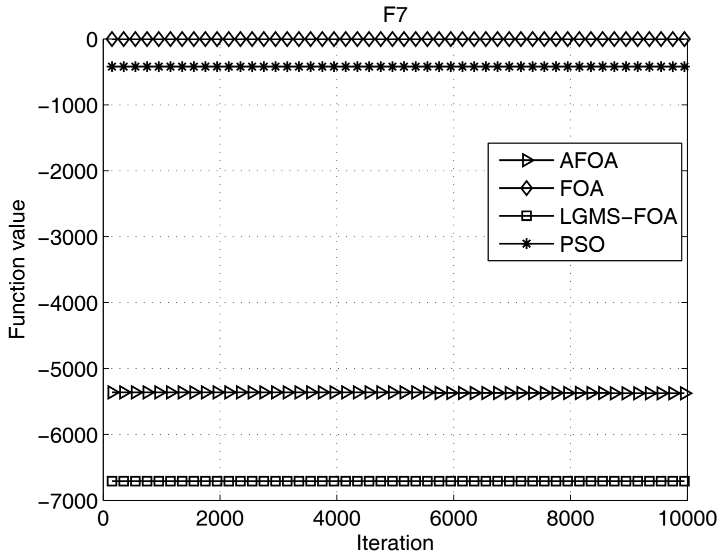

Fig. 9. Comparison of optimization process of F7.

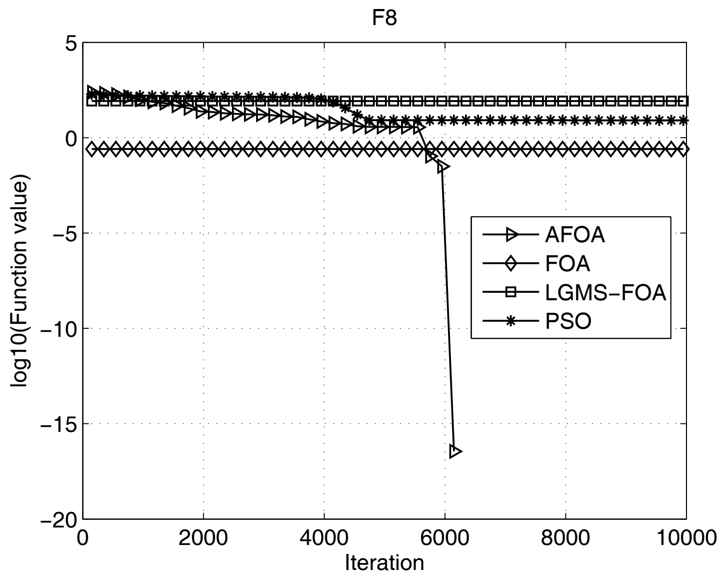

Fig. 10. Comparison of optimization process of F8.

requirement of precision. In Figs. 9 and 12 there are also exist the phenomenons that the minimum is 0 and the curve of AFOA is incomplete. However, it also shows that AFOA has stable global search ability. In Fig. 9, compared with optimization result in Table 1 all of the four contrast functions cannot find the global

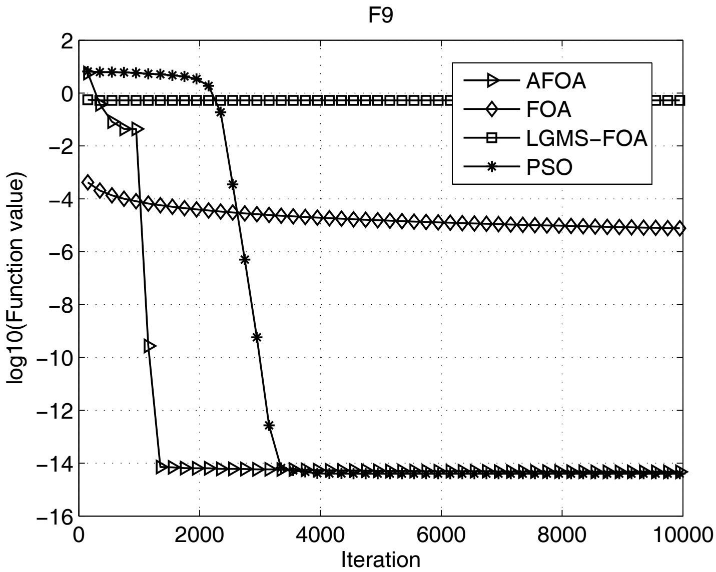

Fig. 11. Comparison of optimization process of F9.

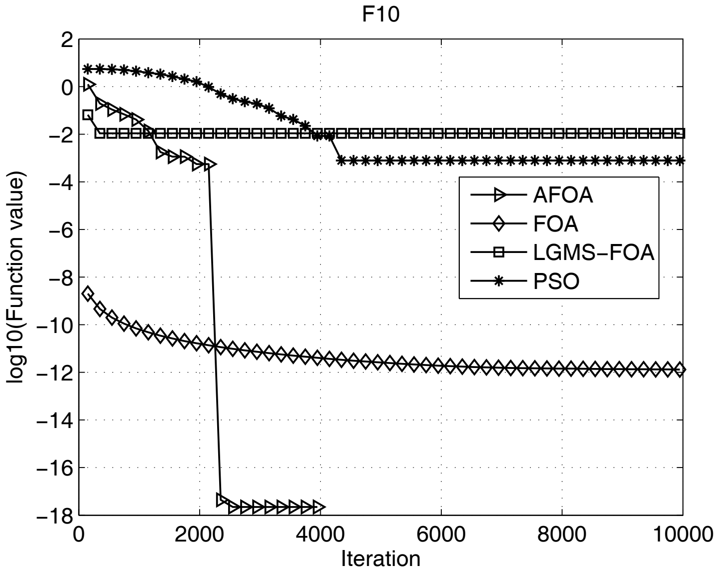

Fig. 12. Comparison of optimization process of F10.

optimum, so we can only compare the gap between them and optimal value. Whereas, Fig. 9 shows that optimization result of AFOA is slightly worse than LGMS-FOA while significantly outperform those of PSO and FOA. Consequently, in the aspect of multidimensional multi-peak function AFOA shows the ascendancy

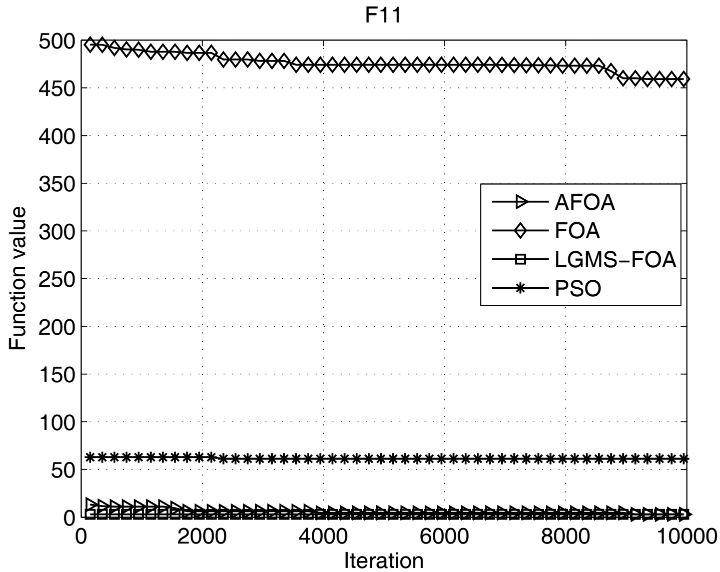

Fig. 13. Comparison of optimization process of F11.

that other three algorithms cannot replace and reflects more stable global optimization capability.

6.3.5. *Results analysis on multidimensional composite multi-peak test function*

The last type is multidimensional composite multi-peak function, which test function $F11$ to $F14$ belongs to. This function tends to be function expression composited by multidimensional variables. Such functions have the characteristics such as multi extremum and constant number of dimensions. The common classical algorithm can also find these extreme points and minimum points, so the major difference is the gap of the result accuracy and the speed of converging to the minimum. As for test function $F11$ to $F14$ in 50 times repeated experiments, the global extremum that is found in each iteration is recorded and the experimental results is shown in Figs. 9–12.

Figures 13, 14 and 16 show that the optimization result of classical FOA is significantly worse than those of AFOA, PSO and LGMS-FOA. The convergence speed of AFOA is slightly slower than LGMS-FOA and PSO, while combined with the optimization results it can be seen that AFOA can find minimum point and has high precision. In Fig. 15, as for test function F13 the initial value of AFOA is significantly worse than the other three algorithms. However, with the number of iterations increasing, the value settles to the global optimal solution ultimately. Thus, overall in the aspect of multidimensional composite multi-peak function, the convergence speed is a little slow, but it still has the global search ability.

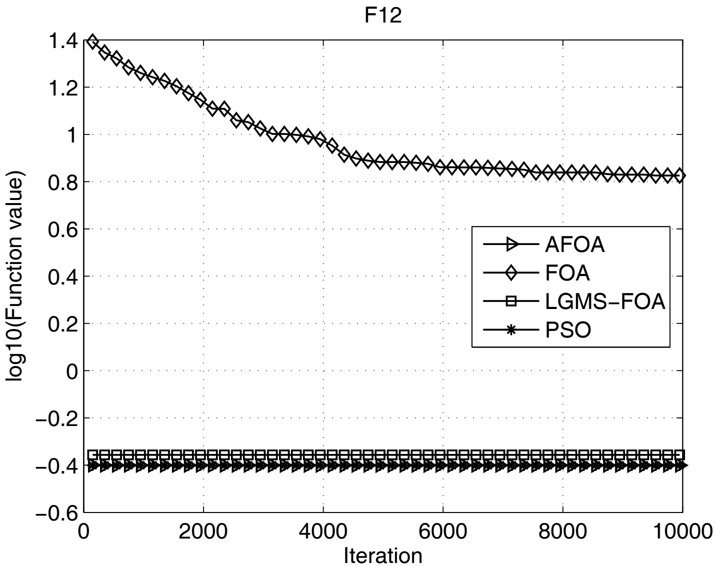

Fig. 14. Comparison of optimization process of F12.

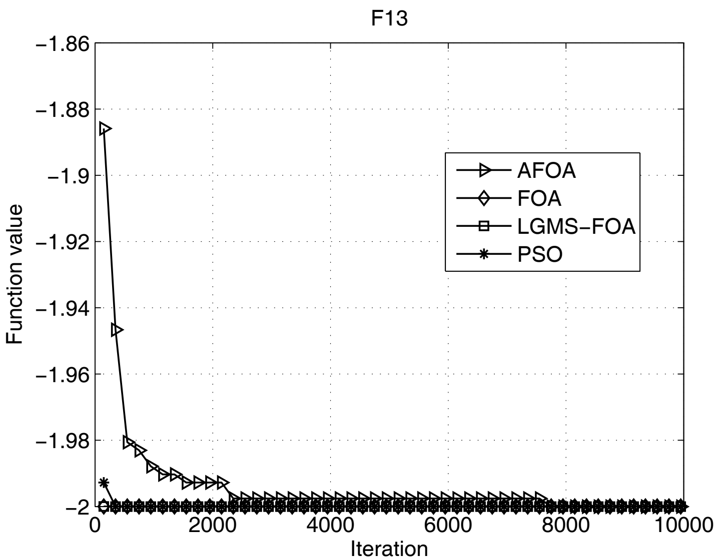

Fig. 15. Comparison of optimization process of F13.

As a whole, from analysis diagrams of experimental results of these five test functions, it infers that when guaranteeing the optimization precision, the convergence speed of AFOA is superior to those of FOA, LGMS-FOA and PSO. Combined with the optimization results, it turns out that the global search ability of AFOA

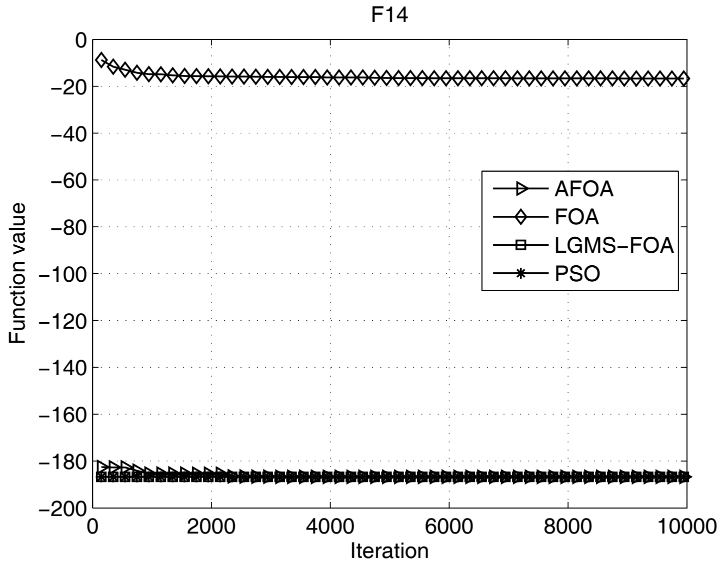

Fig. 16. Comparison of optimization process of F14.

significantly outperforms those of the other three algorithms. Consequently, AFOA is a novel algorithm possessing stable global search capability.

7. Conclusions

This paper puts forward an improved algorithm based on fruit fly optimization algorithm, named as AFOA. AFOA adopts adaptive method to generate swarm range V, avoids falling into local optimum, and effectively strengthens the global search capacity of the algorithm. In the meanwhile, the convergence and the convergence condition of swarm location as well as swarm range V are analyzed. Through the analysis on convergence, the convergence of AFOA is proved theoretically. What's more, through hosts of simulation experiments and comparisons based on the background of five types of different test functions, the global search ability of AFOA is comprehensively verified. Experimental results show that in the testing process of different characteristic functions AFOA can all exhibit stable global search ability, and has faster convergence speed when retaining the optimization precision generally. Consequently, AFOA is an algorithm not only has efficient stable global search capacity but also has stronger universality.

Acknowledgments

This work was supported by the National Key Technology R&D Program (No. 2015BAK24B01), the National Natural Science Foundation of China (No. 61300169), the Natural Science Foundation of Anhui Province of China (No. 1408085MF132) and the 211 Project of Anhui University (No. 02303301).

References

1. M. S. Hossain, M. M. Hassan, M. Al Qurishi and A. Alghamdi, Resource allocation for service composition in cloudbased video surveillance platform, *Proc. of the IEEE Int. Conf. on Multimedia and Expo Workshops* (Melbourne, VIC, 2012), pp. 408–412.
2. H. W. Lin, Y. P. Wang, L. Fan and Y. L. Gao, A new discrete filled function method for finding global minimizer of the integer programming, *Applied Mathematics and Computation* **219** (2013) 4371–4378.
3. V. Uc-Cetina, F. Moo-Mena and R. Hernandez-Ucan, Composition of web services using Markov decision processes and dynamic programming, *The ScientificWorld Journal* (2014) 1–10.
4. V. Gabrel, M. Manouvrier and C. Murat, Optimal and automatic transactional web service composition with dependency graph and 0-1 linear programming, *Service-Oriented Computing*, Vol. 8831 of Lecture Notes in Computer Science (2014), pp. 108–122.
5. M. Chen and Y. H. Yan, QoS-aware service composition over graphplan through graph reachability, *IEEE Int. Conf. on Services Computing, Anchorage* (Alaska, USA, 2014), pp. 544–551.
6. S.-Y. Lin, G.-T. Lin, K.-M. Chao and C.-C. Lo, A cost-effective planning graph approach for large-scale web service composition, *Mathematical Problems in Engineering* (2012) 1–22.
7. S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms* (Springer, 2007).
8. C. S. de Magalhaes, D. M. Almeida, H. J. C. Barbosa and L. E. Dardenne, A dynamic niching genetic algorithm strategy for docking highly flexible ligands, *Information Sciences* **289** (2014) 206–224.
9. Y. Xu, K. Li, J. Hu and K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, *Information Sciences* **270** (2014) 255–287.
10. R. Eberhart and Y. Shi, Particle swarm optimization: Developments, application and resources, in *Proc. of IEEE Congress of Evolutionary Computation* (Seoul, Korea, 2001), pp. 81–86.
11. S. Jamali and V. Shaker, Defense against SYN flooding attacks: A particle swarm optimization approach, *Computers and Electrical Engineering* **40** (2014), 2013–2025.
12. L. Wang *et al.*, Particle swarm optimization based dictionary learning for remote sensing big data, *Knowledge-Based Systems* **79** (2015) 43–50.
13. M. Dorigo, V. Maniezzo and A. Colorni, The ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybernet.* **26** (1996) 29–41.
14. T. J. Liao *et al.*, Ant colony optimization for mixed-variable optimization problems, *IEEE Transaction on Evolutionary Computation* **18**(4) (2014) 503–518.
15. D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005).
16. H. Wang, Z. Wu, S. Rahnamayan, H. Sun, Y. Liu and J. Pan, Multi-strategy ensemble artificial bee colony algorithm, *Information Science* **279** (2014) 587–603.
17. M. S. Kiran, H. Hakli, M. Gunduz and H. Uguz, Artificial bee colony algorithm with variable search strategy for continuous optimization, *Information Science* **300** (2015) 140–157.
18. W. T. Pan, A new fruit fly optimization algorithm: Taking the financial distress model as an example, *Knowledge Based Systems* **26**(2) (2012) 69–74.

19. J.-Q. Li, Q.-K. Pan, K. Mao and P. N. Suganthan, Solving the steelmaking casting problem using an effective fruit fly optimisation algorithm, *Knowledge-Based Systems* **72** (2014) 28–36.
20. H. Dai, G. Zhao, J. Lu and S. Dai, Comment and improvement on "A new fruit fly optimization algorithm: Taking the financial distress model as an example", *Knowledge-Based Systems* **59** (2014) 159–160.
21. A. Azadeh, H. S. Mianaei, S. M. Asadzadeh, M. Saberi and M. Sheikhalishahi, A flexible ANN-GA-multivariate algorithm for assessment and optimization of machinery productivity in complex production units, *Journal of Manufacturing Systems* **35** (2015) 46–75.
22. H. Pourvaziri and B. Naderi, A hybrid multi-population genetic algorithm for the dynamic facility layout problem, *Applied Soft Computing* **24** (2014) 457–469.
23. Y. K. Yi and H. Kim, Agent-based geometry optimization with Genetic Algorithm (GA) for tall apartment's solar right, *Solar Energy* **113** (2015) 236–250.
24. Y. Sun and D. J. (Eric) Verschuur, A Self-adjustable input genetic algorithm for the near-surface problem in geophysics, *IEEE Transaction on Evolutionary Computation* **18**(3) (2014) 309–325.
25. Y. del Valle, G. K. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez and R. G. Harley, Particle swarm optimization: Basic concepts, variants and applications in power systems, *IEEE Transaction on Evolutionary Computation* **12**(2) (2008) 171–195.
26. M. A. M. de Oca, T. Stützle, M. Birattari and M. Dorigo, Frankenstein's PSO: A composite particle swarm optimization algorithm, *IEEE Transactions on Evolutionary Computation* **13**(5) (2009) 1120–1132.
27. S. Alam, G. Dobbie, P. Riddle and Y. S. Koh, Hierarchical PSO clustering based recommender system, *IEEE World Congress on Computational Intelligence* (*WCCI 2012*) (Brisbane, Australia, 2012), pp. 10–15.
28. W. Cheng, B. Scheuermann and M. Middendorf, Quick-ACO: Accelerating ant decisions and pheromone updates in ACO, *Evolutionary Computation in Combinatorial Optimization*, Vol. 6622 of Lecture Notes in Computer Science (2011), pp. 238–249.
29. G. Leguizamón and C. A. C. Coello, A boundary search based ACO algorithm coupled with stochastic ranking, *2007 IEEE Congress on Evolutionary Computation* (*CEC 2007*) (Singapore, 2007), pp. 165–172.
30. Q. Yu, L. Chen and B. Li, Ant colony optimization applied to web service compositions in cloud computing, *Computers and Electrical Engineering* **41** (2015) 18–27.
31. W. T. Pan, A new evolutionary computation approach: Fruit fly optimization algorithm, in *2011 Conference of Digital Technology and Innovation Management* (Taipei). Program code on the website: http://www.oitecshop.byethost16.com/FOA.html, (2011).
32. W. T. Pan, *Fruit Fly Optimization Algorithm* (Tsang Hai Book Publishing Co., Taipei, 2011).
33. S. M. Lin, Analysis of service satisfaction in web auction logistics service using a combination of fruit fly optimization algorithm and general regression neural network, *Neural Computing and Applications* **22** (2013) 783–791.
34. E. Rashedi, H. Nezamabadi-pour and S. Saryazdi, GSA: A gravitational search algorithm, *Information Science* **179** (2009) 2232–2248.
35. S. Moein and R. Logeswaran, KGMO: A swarm optimization algorithm based on the kinetic energy of gas molecules, *Information Science* **275** (2014) 127–144.

36. D. Shan, G. H. Cao and H. J. Dong, LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems, *Mathematical Problems in Engineering* (2013) 1–9.
37. J. Kennedy and R. C. Eberhart, Particle swarm optimization, in *Proc. of IEEE Int. Conf. on Neural Networks*, Vol. 4 (1995), pp. 1942–1948.