Journal of Computational Science 23 (2017) 226–239

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

A new meta-heuristic butterfly-inspired algorithm

Xiangbo Qi [a,b,*], Yunlong Zhu [a], Hao Zhang [a]

[a] Shenyang Institute of Automation Chinese Academy of Sciences, 110016, Shenyang, China
[b] University of Chinese Academy of Sciences, 100049, Beijing, China

ARTICLE INFO

Article history:
Received 14 July 2016
Received in revised form 3 April 2017
Accepted 5 June 2017
Available online 8 June 2017

Keywords:
Artificial butterfly optimization
Artificial bee colony algorithm
Particle swarm optimization
Genetic algorithm

ABSTRACT

This paper proposes a novel bio-inspired algorithm named Artificial Butterfly Optimization (ABO) algorithm. The new algorithm is based on the mate-finding strategy of some butterfly species. Two groups of artificial butterflies are employed for simulating the flight strategies. If the flight strategies of artificial butterflies are redefined, ABO can develop a new algorithm. From this point, ABO is a mimic-life algorithm in grandness. By presenting three flight strategies, we build two new algorithms named ABO1 and ABO2. We validate the two new algorithms and compare their performance with other well-known nature-inspired algorithms on twenty-two benchmark functions. The experimental results show that the proposed algorithm is able to provide very promising and competitive results on most benchmark functions. It also proves that the ABO algorithm provides a new effective computational framework for solving optimization problems.

1. Introduction

Many metaheuristic algorithms have taken inspiration from biological systems in nature. For example, Particle swarm optimization (PSO) algorithm was inspired by the swarm behavior of birds and fish [1]. Firefly algorithm (FA) was inspired by the bioluminescent communication behavior of fireflies [2]. Artificial Fish Swarm Algorithm (AFSA) was inspired by prey, swarm and follow behavior of a school of fish [3]. Artificial bee colony (ABC) algorithm was inspired by the foraging behaviors of honey bee colony [4]. Genetic Algorithm (GA) was inspired by the process of natural evolution [5]. Bacterial foraging optimization (BFO) was inspired by the foraging behavior of bacteria [6]. Plant growth simulation algorithm (PGSA) was inspired by plant growth [7]. These algorithms have been used to solve many tough optimization problems. However, "No free Lunch" theorems [8,9] suggest that one algorithm impossibly shows the best performance for all problems. Every algorithm has its advantages and disadvantages. For example, with increasing of the dimensionality of the search space, many stochastic optimization algorithms such as ABC, PSO and GA, have a poor convergence behavior. Many strategies including improving existing algorithms or studying new algorithms can get better optimization effects. Generally, a well designed algorithm can do well in balancing two key processes: exploration and exploitation. Exploitation intends to search carefully and converge to the optimum while exploration makes an algorithm escape from the local optimum. Though some animals are unsophisticated individually, they make wonders as a group by some interaction with each other. An answer is the power of natural selection over millions of years. Inspired by the mate-finding strategy of speckled woods (a woodland butterfly), this paper proposes a new algorithm named Artificial Butterfly Optimization (ABO) algorithm.

The remainder of the article is organized as follows. Section 2 introduces the mate-finding behavior of the male speckled woods. Section 3 proposes a novel artificial butterfly optimization algorithm, gives the pseudo code and develops two new algorithms by using different flight strategies in ABO algorithm. Section 4 gives 2-Dimensional simulations of the new algorithm on two well-known test functions and discusses parameter tuning in different ABO algorithms. This section also gives the experiment process of comparison with other algorithms in detail, presents the experimental results and gives the analysis. Finally, Section 5 gives the conclusions.

2. Behavior of speckled woods

Many male animals defend territories as a means of monopolizing receptive females and maximizing their number of matings [10]. Butterflies are no exception. Different butterfly species show different behaviors. The speckled wood (Pararge aegeria) is a wood-

* Corresponding author at: Dept. of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, 110016, Shenyang, China.
E-mail addresses: qixb@sia.cn, drqixiangbo@gmail.com, ustcdragon@126.com (X. Qi).

http://dx.doi.org/10.1016/j.jocs.2017.06.003

Fig. 1. The behavior of Speckled Woods contending for sunspots.

land butterfly that has a rather complicated life history [11,12]. This butterfly is found in and on the borders of woodlands. In the woodland, lots of sunlit patches (called sunspots in this paper) are formed on the ground when the sun shines on the trees. It seems likely that the sunny spots are preferred by speckled woods because it is here that there is sufficient warmth to enable the butterflies to remain active [14]. Many factors makes a forest sunlit patch a rendezvous site [13]. In regard to speckled woods' mate-finding, two alternative male strategies have been described: dominant males adopt a perching strategy monopolizing large sunspots on the forest floor, and subdominant males adopt a patrolling strategy [14]. N. B. Davies characterized the mate-finding strategy of male speckled woods by contending for sunspots [15]:

1) Male speckled woods competed for territories. The territories referred to the spots of sunlight on the ground layer of woodland, which are the best places for finding females.
2) At the same time only 60% of the male speckled woods called sunspot butterflies in this paper occupied sunspots; the remainder called canopy butterflies in this paper patrolled for females up in the tree canopy.
3) Canopy butterflies continually flew down from the canopy and attempted to occupy one sunspot. If the sunspot was already occupied, a spiral flight took place between both butterflies. Finally, one butterfly flew towards the tree canopy and the other one occupied the sunspot.
4) A male speckled wood wanted to occupy a bigger sunspot because the larger the sunspot, the more females visit it.

Fig. 1 is an example describing the behavior of speckled woods contending for sunspots. The sunspot butterflies have taken better locations. Furthermore, each sunspot butterfly wants to take a larger sunspot. The canopy butterflies intending to take a better location always attempt to fly towards a sunspot butterfly. If a canopy butterfly wins the sunspot, it becomes a sunspot butterfly. If the canopy butterfly doesn't win the sunspot, it flies towards any other direction.

3. Artificial butterfly algorithm

Butterfly territoriality offers an excellent model system for specific investigation into how life history optimization affects lifetime investment in aggressive behavior [16]. Speckled woods also offer a wonderful inspiration for proposing a new optimization algorithm.

Table 1
Pseudo code of ABO algorithm.

1. Initialize the locations of butterfly population
2. Evaluate the fitness of every butterfly
3. While not meet the terminal condition
4. Sort all butterflies by their fitness
5. Select some butterflies with better fitness to form sunspot butterflies, the rest form canopy butterflies
6. For each sunspot butterfly
fly to one new location according to **sunspot flight mode**
Evaluate the fitness of the new sunspot
apply greedy selection on the original location and the new one
End for
7. For each canopy butterfly
Fly to one randomly selected sunspot butterfly according to **canopy flight mode**
Evaluate the fitness
8. If better fitness
Apply greedy selection on the original location and the new one
9. else
Fly to new location according to **free flight mode**
End if
10. End for
11. End while
12. Postprocess results

Based on the mate-finding strategy of speckled woods, we proposed the Artificial Butterfly Optimization (ABO) algorithm. The pseudo code of ABO algorithm is listed in Table 1. Some rules are made to idealize the mate-finding strategies of butterflies in ABO algorithm:

(a) In order to increase the likeliness of encountering female butterflies, all male butterflies attempt to fly towards a better location called a sunspot.
(b) To occupy a better sunspot, each sunspot butterfly always attempts to fly to its neighbor's sunspot.
(c) Each canopy butterfly continually flies towards any sunspot butterfly to contend for the sunspot.

As can be seen from Table 1, the original butterfly population was divided two groups by their fitness. Different flight strategy is applied to the each group. In this point, it is somewhat similar to niching techniques. A niching method [17,18] is usually used modify the behavior of a classical algorithm to maintain various groups in the adopted population component in order to effectively find multiple optima. There are three flight modes including sunspot flight mode, canopy flight mode and free flight mode. These modes can be given different flight strategies. That is to say, ABO can produce a new algorithm when any one of three flight modes is given a new flight strategy.

Here, we give three flight strategies for a virtual butterfly. It is worth noting that this paper utilizes a D-dimensional vector to represent the location of a virtual butterfly.

1) Each butterfly flies towards a randomly selected neighbor following Eq. (1). This strategy is used for the sunspot flight mode or the canopy flight mode in the ABO algorithm.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + (X_{i,j}^{t} - X_{k,j}^{t}) \cdot rand() \tag{1}$$

where i is the ith butterfly. j is a randomly selected dimension index between [1,D], t is the number of iterations, $rand()$ generates a random number between [-1,1], and k is a randomly selected butterfly. Here, k is different from i.

2) Each butterfly flies towards a randomly selected neighbor one following Eq. (2). This strategy is used for the sunspot flight mode or the canopy flight mode in ABO algorithm.

$$X_i^{t+1} = X_i^t + \frac{X_k^t - X_i^t}{\|X_k^t - X_i^t\|} \cdot (Ub - Lb) \cdot step \cdot rand() \tag{2}$$

where i is the ith virtual butterfly, t is the number of iterations, $step$ is the new location of the ith virtual butterfly, $step$ is the flight distance, $rand()$ generates a random number between (0,1) and k is a randomly selected butterfly. Here, k is different from i.Lb is the lower boundary value of the flying range for the ith virtual butterfly and Ub is the upper boundary value of the flying range of the ith virtual butterfly. Lb and Ub are relevant to a specific problem.

3) Each butterfly flies towards a randomly selected neighbor following Eq. (3). The same approach has been utilized to search for a new position in the exploration phase [19]. This strategy is used for the free flight mode in ABO algorithm.

$$X_i^{t+1} = X_k^t - 2 \cdot a \cdot rand() - a \cdot D \tag{3}$$

where i is the ith virtual butterfly, X_i^{t+1} is the new location of the ith virtual butterfly, a is linearly decreased from 2 to 0 over the course of iteration and $rand()$ generates a random number between (0,1). k is a randomly selected butterfly. D is a randomly produced value following Eq. (4).

$$D = |2 \cdot rand() \cdot X_k^t - X_i^t| \tag{4}$$

where i is the virtual butterfly, k is a randomly selected butterfly, $rand()$ generates a random number between (0,1).

The parameter step in Eq. (2) is not a fixed value. A decreasing strategy according to Eq. (5) is used. Step decreases from 1 to $step_e$ linearly with iterations. In the early stage, larger step value provides better global searching ability and diversity. In the late stage, smaller step value avoids the large jumping and provides better convergence. The strategy could keep balance between the exploration and exploitation abilities of the proposed algorithm.

$$step = 1 - (1 - step_e) \cdot \frac{E}{\max E} \tag{5}$$

where E is current evaluations count and max E is the max evaluations count.

In this paper, we build a new algorithm called ABO1 when both the sunspot flight mode and the canopy flight mode in ABO use the first flight strategy following Eq. (1). We also build a new algorithm called ABO2 when both the sunspot flight mode and the canopy flight mode in ABO use the second flight strategy following Eq. (2).

4. Validation and comparison

For the ease of visualization, we have implemented all algorithms using Matlab for various test functions. In order to compare the different algorithms fairly, we use a number of function evaluations (FEs) as a measure criterion in this paper.

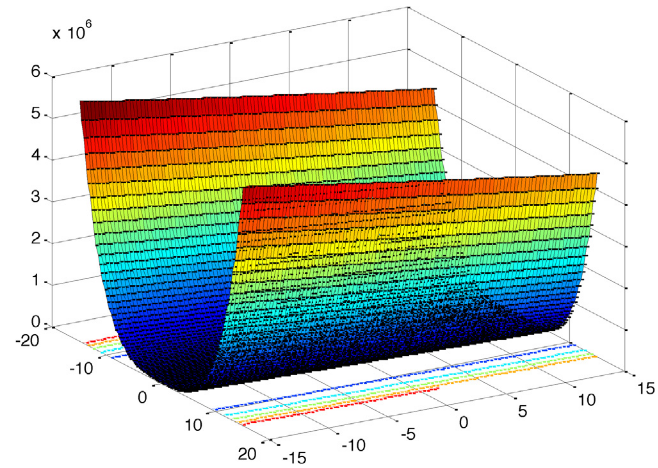


Fig. 2. The surface of 3D-Rosenbrock function.

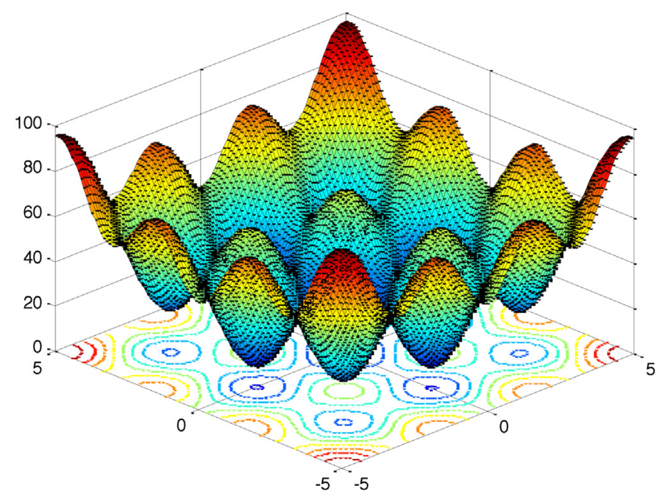


Fig. 3. The surface of 3D-Eggcrate function.

4.1. Validation and parameter study

In the current benchmark validation, we chose two well-known benchmark functions including the Rosenbrock function and the Eggcrate function. The first function, the RosenBrock function, has a narrow valley near the global optimum. So it is very difficult to converge to the global optimum. The surface of 3D-Rosenbrock's function is shown in Fig. 2. The second function, the Eggcrate function, is a multimodal function. The surface of 3D-Eggcrate function is shown in Fig. 3.

Rosenbrock function:

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2, (x, y) \in [-15, 15] \times [-15, 15]$$

$f(x, y)$ has a global minimum 0 at (1,1).

The eggcrate function:

$$g(x, y) = x^2 + y^2 + 25(\sin^2 x + \sin^2 y), (x, y) \in [-2\pi, 2\pi] \times [-2\pi, 2\pi]$$

$g(x, y)$ has a global minimum 0 at(0,0).

In our implementation, we use 40 virtual butterflies in ABO1. For the Rosenbrock 2-D banana function, two snap shots of the loca-

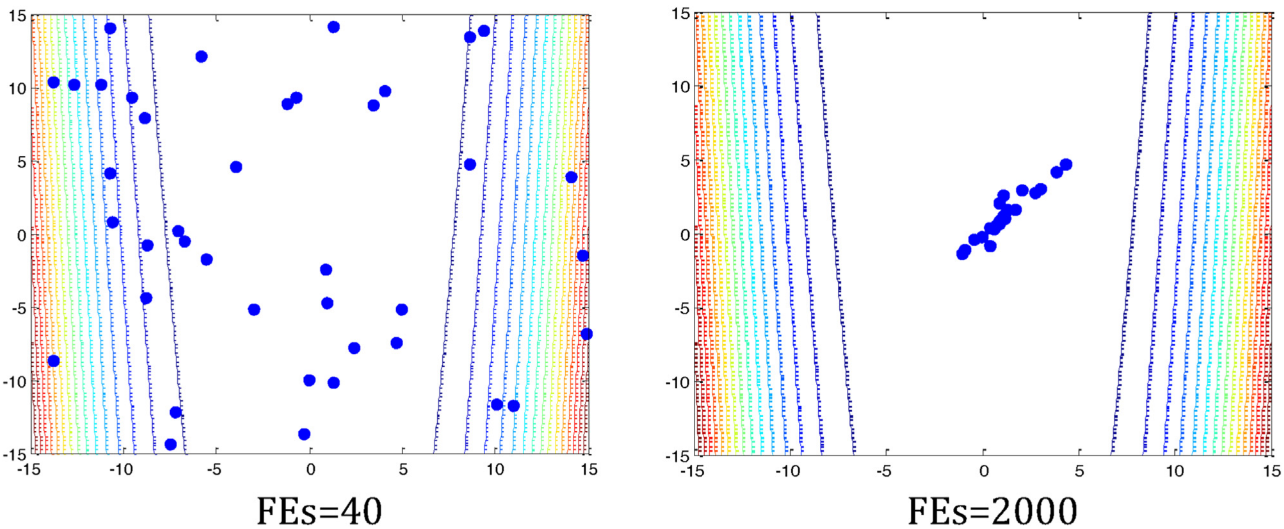


Fig. 4. The locations of 40 butterflies in the Rosenbrock function.

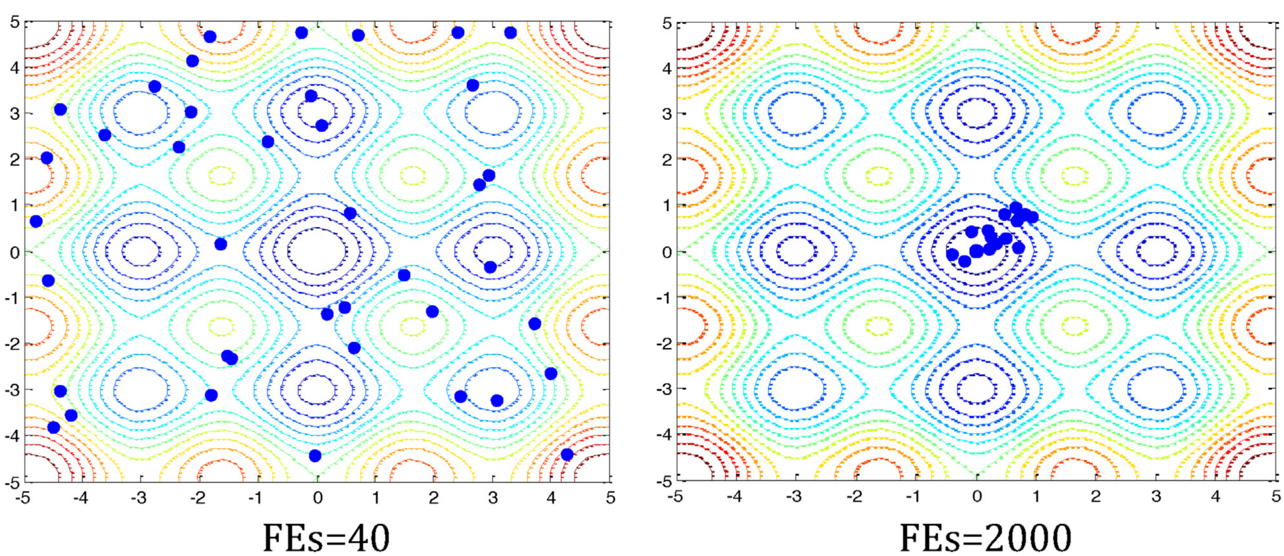


Fig. 5. The locations of 40 butterflies in the Eggcrate function.

tions of 40 virtual butterflies are shown in Fig. 4. From the figure, most virtual butterflies fly towards the global optimum (1, 1). For the Eggcrate function, two snapshots of the locations of 40 virtual butterflies are shown in Fig. 5. Similarly, the virtual butterflies fly towards the global optimum (0, 0).

We have also tried to vary the number of the population size (n). We used n = 10, 20, 30, 40, 80, 100, and found n = 20 is sufficient for most optimization problems.

The proportion of sunspot butterflies in ABO1 algorithm is not a fixed value. The proportion of sunspot butterflies is linearly decreased from 0.9 to $ratio_e$ over the course of iteration. Six continuous benchmark functions, Sphere 30D, Quadric 30D, Sinproblem 30D, Rastrigin 30D,Ackley 30D and Griewank 30D are employed to investigate the impact of this parameter. Set $ratio_e$ equal to different value and all the functions run 30 sample times. The experimental results in terms of mean values and standard deviation of the optimal solutions over 30 runs were listed in Table 2. From Table 2, we can find that ABO1 with $ratio_e$ equal to 0.2 performs best on three functions among all six functions. According to the results with different $ratio_e$ values, we chose $ratio_e$ equal to 0.2 as an optimal value for the next experiments.

$step_e$ is a parameter in the ABO2 algorithm. Six continuous benchmark functions, Sphere 30D, Quadric 30D, Sinproblem 30D, Rastrigin 30D,Ackley 30D and Griewank 30D are employed to investigate the impact of this parameter. Set $step_e$ equal to different value and all the functions run 30 sample times. The experimental results in terms of mean values and standard deviation of the optimal solutions over 30 runs were listed in Table 3. From Table 3, we can find that ABO2 with $step_e$ equal to 0.05 performs best on four functions among all six functions. According to the results with different $step_e$ values, we chose $step_e$ equal to 0.05 as an optimal value for the next experiments.

4.2. Comparison with other algorithms

In order to compare the performance of ABO1 and ABO2, artificial bee colony algorithm (ABC) [4], canonical PSO with constriction factor (PSO) [20], cooperative PSO (CPSO) [21] and standard genetic algorithm (GA) [22] were employed for comparison. ABC and PSO are classical population-based paradigms simulating foraging behavior of social animals. CPSO is a cooperative PSO model, cooperatively coevolving multiple PSO subpopulations. GA is the

Table 2
Results of ABO1 on six benchmarks with different . In bold are the best results.

Function		0.8	0.6	0.4	0.2	0.1
Sphere	Mean	1.121180e-016	9.107661e-017	7.593313e-017	**6.853472e-017**	8.620071e-017
	Std	6.782169e-017	4.947150e-017	4.664341e-017	**2.924991e-017**	5.532235e-017
Quadric	Mean	4.901760e+001	4.305358e+001	2.310944e+001	**2.232625e+001**	2.651196e+001
	Std	1.781078e+001	2.426605e+001	2.100164e+001	**1.885170e+001**	1.545503e+001
Sinproblem	Mean	6.385729e-016	**6.122152e-016**	6.839769e-016	6.972564e-016	7.340031e-016
	Std	1.042064e-016	**1.156867e-016**	1.223919e-016	1.256198e-016	1.528191e-016
Rastrigin	Mean	3.316530e-002	**0.000000e+000**	**0.000000e+000**	**0.000000e+000**	**0.000000e+000**
	Std	1.816538e-001	**0.000000e+000**	**0.000000e+000**	**0.000000e+000**	**0.000000e+000**
Ackley	Mean	7.993606e-015	6.927792e-015	6.572520e-015	6.454097e-015	**5.980401e-015**
	Std	0.000000e+000	1.655890e-015	1.770221e-015	1.790592e-015	**1.790592e-015**
Griewank	Mean	**2.465347e-004**	3.285762e-004	4.106996e-004	4.105358e-004	7.391119e-004
	Std	**1.350326e-003**	1.799686e-003	2.249495e-003	2.248597e-003	2.831249e-003

Table 3
Results of ABO2 on six benchmarks with different $step_e$. In bold are the best results.

Function		0.5	0.1	0.05	0.01	0.005
Sphere	Mean	8.683040e-017	8.052833e-017	**7.893010e-017**	8.333587e-017	9.005905e-017
	Std	5.479654e-017	4.078720e-017	**4.483304e-017**	4.384351e-017	6.388527e-017
Quadric	Mean	1.267400e+002	8.106600e+001	6.431082e+001	**5.493870e+001**	7.164418e+001
	Std	6.401104e+001	6.920753e+001	3.876923e+001	**3.464182e+001**	7.594913e+001
Sinproblem	Mean	3.274431e-001	3.760775e-001	**2.822156e-001**	3.107061e-001	3.195262e-001
	Std	2.827977e-001	2.423145e-001	**2.156319e-001**	3.049761e-001	2.015661e-001
Rastrigin	Mean	1.155074e-001	1.567831e-006	**7.579123e-015**	1.080025e-013	1.136868e-014
	Std	6.326384e-001	8.587362e-006	**2.467985e-014**	3.987012e-013	2.312607e-014
Ackley	Mean	6.217249e-015	**5.625130e-015**	6.572520e-015	5.861978e-015	6.098825e-015
	Std	1.806724e-015	**1.703396e-015**	1.770221e-015	1.770221e-015	1.802705e-015
Griewank	Mean	4.849652e-003	5.054321e-003	**3.101092e-003**	3.224361e-003	5.194855e-003
	Std	1.497205e-002	1.559055e-002	**1.182118e-002**	1.229274e-002	1.606284e-002

classical stochastic search technique mimicking the process of natural selection and is undoubtedly the most popular stochastic optimization algorithm. In addition, a set of twenty-two well-known benchmark functions were used in this experiment.

4.2.1. Benchmark functions

The twenty-two benchmark functions are widely adopted by other researchers to test their algorithms in many works [23–25]. In this paper, all functions used their standard ranges. These benchmark functions totaling twenty-two diverse and difficult minimization problems comprise six unimodal continuous functions (f_1-f_6), eight multimodal continuous functions (f_7-f_{14}) and eight composition function (f_{15}-f_{22}). The formulas of these functions are presented in Tables 4 and 5. Functions f_{11}-f_{14} are four rotated functions employed in Liang's work [26]. In the rotated functions, a rotated variable y, which is produced by the original variable x left multiplied an orthogonal matrix, is used to calculate the fitness (instead of x). The orthogonal matrix is generated according to Salomon's method [27]. The composition functions were specifically designed for the competition and comprise the sum of three of five unimodal and/or multimodal functions, leading to very challenging properties: multimodal, non-separable, asymmetrical and with different properties around different local optima.

4.2.2. Experiment sets

The population size of all algorithms was 20. The maximum evaluation count on dimensions 30 is 100,000. In order to do meaningful statistical analysis, each algorithm runs for 30 times and takes the mean value and the standard deviation value as the final result. For CPSO and PSO, the learning rates *C1* and *C2* were both set as 2.05. The constriction factor $X = 0.729$. The split factor for CPSO is equal to the dimensions. In GA, single-point crossover is employed. The probability of crossover is 0.95 and the probability of mutation is 0.1.

All algorithms were implemented in Matlab R2010a using computer with Intel Core i5–2450 M CPU, 2.5 GHz, 2GB RAM. The operating system of the computer is Windows7.

4.2.3. Experiment results and analysis

The experimental results, including the mean and the standard deviation of the function values obtained by the six algorithms with 30 dimensions, are listed in Table 6. Best values obtained on each function are marked as bold. Rank represents the performance order of the six algorithms on each benchmark function. It is obvious that the ABO1 algorithm performed best on most functions.

The mean best function value profiles of the twenty-two algorithms with 30 dimensions are shown in Fig. 6.

1) Continuous unimodal functions

As can be seen in Table 6, on Sphere function, the performance order for the six intelligent algorithms is ABO1>ABO2>ABC>CPSO>PSO>GA. ABO1, ABO2 and ABC got satisfactory results. As can be seen from Fig. 6 (f_1), ABC converges fast at first, but traps in the local minimum after about 20, 000 FEs. Finally, ABO1 and ABO2 obtain better values than ABC in terms of accuracy and robustness.

On Powers function, the performance order for the six intelligent algorithms is ABO1>ABO2>PSO>ABC>CPSO>GA. All algorithms obtained acceptable results except the GA. ABO1 and ABO2 perform better than the other algorithms in terms of convergence speed, seen in Fig. 6 (f_2).

On Sumsquares function, the performance order for the six intelligent algorithms is ABO1>ABO2>ABC>CPSO>PSO>GA. The performance of six algorithms is much similar with the Sphere function, seen in Fig. 6(f_1) and (f_3).

On Quadric function, the performance order for the six intelligent algorithms is PSO>ABO1>CPSO>ABC>ABO2>GA. As seen in Fig. 6 (f_4), GA hardly converge. PSO performs best. ABO1, ABO2, ABC and CPSO converged slowly.

On Sin problem function, the performance order for the six intelligent algorithms is ABO1>ABC>CPSO>ABO2>GA>PSO. PSO, GA

Table 4
Classical test suite.

Name	Function	Limits
Sphere(f_1)	$f_1 = \sum_{i=1}^{D} {x_i}^2$	$[-5.12, 5.12]^D$
powers(f_2)	$f_2 = \sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	$[-1, 1]^D$
Sumsquares(f_3)	$f_3 = \sum_{i=1}^{D} ix_i^2$	$[-10, 10]^D$
Quadric(f_4)	$f_4 = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$	$[-10, 10]^D$
Sinproblem(f_5)	$f_5 = \frac{\pi}{D}\left\{10\sin^2 \pi x_1 + \sum_{i=1}^{D-1}(x_i-1)^2(1+10\sin^2\pi x_{i+1}) + (x_D-1)^2\right\}$	$[-10, 10]^D$
zakharov(f_6)	$f_6 = \sum_{i=1}^{D} x_i^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{D} 0.5ix_i\right)^4$	$[-5, 10]^D$
rastrigin(f_7)	$f_7 = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-15, 15]^D$
schwefel(f_8)	$f_8 = D \cdot 418.9829 + \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i\rvert})$	$[-500, 500]^D$
ackley(f_9)	$f_9 = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right)$	$[-32.768, 32.768]^D$
griewank(f_{10})	$f_{10} = \frac{1}{4000}(\sum_{i=1}^{D} x_i^2) - (\prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}})) + 1$	$[-600, 600]^D$
rot_rastrigin(f_{11})	$f_{11} = f_7(y), y = M \times x$	$[-15, 15]^D$
rot_schwefel(f_{12})	$f_{12} = f_8(y), y = M \times x$	$[-500, 500]^D$
rot_ackley(f_{13})	$f_{13} = f_9(y), y = M \times x$	$[-32.768, 32.768]^D$
rot_griewank(f_{14})	$f_{14} = f_{10}(y), y = M \times x$	$[-600, 600]^D$

Table 5
CEC 2013 test suite.

Name	Limits
Composition Function 1 (n=5,Rotated) (f_{15})	$[-100, 100]^D$
Composition Function 2 (n=3,Unrotated) (f_{16})	$[-100, 100]^D$
Composition Function 3 (n=3,Rotated) (f_{17})	$[-100, 100]^D$
Composition Function 4 (n=3,Rotated) (f_{18})	$[-100, 100]^D$
Composition Function 5 (n=3,Rotated) (f_{19})	$[-100, 100]^D$
Composition Function 6 (n=5,Rotated) (f_{20})	$[-100, 100]^D$
Composition Function 7 (n=5,Rotated) (f_{21})	$[-100, 100]^D$
Composition Function 8 (n=5,Rotated) (f_{22})	$[-100, 100]^D$

and ABO2 hardly converge. CPSO converged slowly. ABC and ABO1 got satisfactory results. ABC trapped in a local minimum at about 30,000FEs. ABO1 trapped in a local minimum at about 60,000FEs. ABO1 got the best result.

On Zakharov function, the performance order for the six intelligent algorithms is ABO2>ABO1>PSO>CPSO>ABC>GA. Before about 10000 FEs, ABC and GA are trapped in the local minimum, seen in Fig. 6 (f_6). ABO1 and ABO2 perform better.

From the comparisons between ABO1, ABO2 and the other algorithms, we can see that ABO1 and ABO2 have better performances on continuous unimodal functions f_1,f_2,f_3 and f_6.

2) Continuous multimodal functions

The multimodal functions f_{11}-$f_{14}$are regarded as the most difficult functions to optimize since the number of local minima increases exponentially as the function dimension increases. The test functions f_{11}-f_{14} are four rotated functions. The rotated functions are much more difficult to solve.

On Rastrigin function, the performance order for the six intelligent algorithms is ABO1>ABO2>ABC>PSO>CPSO>GA. ABO2 converged very slow at the very beginning, and converged fast at about 80,000FEs, then trapped in a local minimum at about 90,000FEs. At last, ABO2 got a good result. ABC converged slower at first, and then converged fast at about 70,000FEs. At about 80,000FEs, ABC trapped in a local minimum. ABO1 performs best and the result was improved continually, seen in Fig. 6 (f_7).

On Schwefel function, the performance order for the six intelligent algorithms is ABC>CPSO>ABO1>ABO2>GA>PSO. As seen in Fig. 6 (f_8), GA and PSO hardly converge. ABO1 perform much better than ABO2, GA and PSO in terms of accuracy, robustness, and convergence speed. CPSO converged very fast at the very beginning, and trapped in a local minimum at about 10,000FEs. ABC performs best.

On Ackley function, the performance order for the six intelligent algorithms is ABo1>ABO2>ABC>CPSO>PSO>GA. PSO and GA hardly converged. CPSO converged slowly. The results for ABC, ABO1 and ABO2 were improved continually. The convergence speed of ABC is better than the one of ABO1 and ABO2.However, ABC trapped in a local minimum at about 50,000FEs. ABO1 and ABO2 trapped in a local minimum at about 60,000FEs, seen in Fig. 6 (f_9).

On Griewank function, the performance order for the six intelligent algorithms is ABO1>ABC>ABO2>CPSO>PSO>GA. PSO, CPSO, ABC and ABO2 converged fast at first, and then converged very slowly. The result of ABO1 is improved continually and trapped in a local minimum at about 30,000FEs, seen in Fig. 6 (f_{10}).

On Rotated Rastrigin function, the performance order for the six intelligent algorithms is ABO2>ABO1>ABC>PSO>CPSO>GA. On Rotated Rastrigin function, ABO2 is 1 order of magnitude better

Table 6
Results comparison of different optimal algorithms with dimension of 30 .

Function		ABO1	ABO2	ABC	PSO	CPSO	GA
f_1	Mean	**6.698830e-017**	7.155941e-017	8.052865e-016	2.219698e-004	9.718608e-006	2.168373e+000
	Std	**3.101193e-017**	3.884028e-017	1.277349e-016	1.391320e-004	5.504014e-006	8.592691e-001
	Best	3.740350e-017	4.545520e-017	6.033220e-016	4.728400e-005	2.984920e-006	9.105550e-001
	Worst	2.198343e-016	2.141320e-016	1.204110e-015	6.932630e-004	2.586070e-005	4.622420e+000
	Rank	1	2	3	5	4	6
f_2	Mean	**2.007439e-017**	2.181664e-017	1.729907e-014	1.631280e-014	5.694421e-009	1.142725e-002
	Std	**1.118322e-017**	1.008276e-017	2.826866e-014	6.192908e-014	1.613811e-008	5.011915e-003
	Best	1.652855e-018	8.807930e-018	6.785320e-017	3.466710e-018	2.389930e-013	3.433350e-003
	Worst	4.785918e-017	5.155240e-017	1.201480e-013	3.364940e-013	8.705580e-008	2.441760e-002
	Rank	1	2	4	3	5	6
f_3	Mean	**7.617745e-017**	9.584096e-017	7.947345e-016	4.251270e-002	5.624401e-004	7.265310e+001
	Std	**4.043972e-017**	6.050023e-017	1.230775e-016	5.274829e-002	2.982597e-004	3.432228e+001
	Best	4.028703e-017	3.623430e-017	4.038980e-016	7.852210e-003	2.260130e-004	3.165710e+001
	Worst	2.137263e-016	2.431550e-016	9.856360e-016	2.805710e-001	1.481230e-003	1.842400e+002
	Rank	1	2	3	5	4	6
f_4	Mean	2.154363e+001	6.350493e+001	5.032977e+001	**3.493495e+000**	4.015971e+001	2.336045e+002
	Std	1.232025e+001	3.379331e+001	1.340306e+001	**1.268382e+001**	4.875644e+001	5.392350e+001
	Best	5.832321e+000	2.336160e+001	2.663750e+001	5.709330e-002	1.211640e+000	1.493470e+002
	Worst	5.466066e+001	1.735600e+002	8.050400e+001	5.021520e+001	1.526090e+002	3.520610e+002
	Rank	2	5	4	1	3	6
f_5	Mean	**7.515154e-016**	3.033825e-001	7.608140e-016	4.450671e+000	1.041101e-005	3.240375e+000
	Std	**1.659126e-016**	2.152767e-001	1.554742e-016	3.244239e+000	1.595031e-005	1.528852e+000
	Best	5.035031e-016	3.170060e-003	4.744540e-016	2.912530e-001	1.646270e-006	1.252960e+000
	Worst	1.177538e-015	8.069690e-001	1.185790e-015	1.639490e+001	8.254270e-005	6.867520e+000
	Rank	1	4	2	6	3	5
f_6	Mean	2.256266e+001	**1.987342e+000**	2.194447e+002	1.338578e+002	2.106311e+002	4.302965e+002
	Std	8.799411e+000	**1.775732e+000**	3.597426e+001	1.219247e+002	8.063483e+001	9.529330e+001
	Best	6.474196e+000	5.430510e-001	1.639170e+002	7.990490e-002	6.958440e+001	2.648770e+002
	Worst	4.179764e+001	8.002880e+000	2.865630e+002	5.000310e+002	4.040080e+002	6.260390e+002
	Rank	2	1	5	3	4	6
f_7	Mean	**0.000000e+000**	4.547474e-014	4.069932e-010	1.243182e-009	1.442855e-002	2.914067e+002
	Std	**0.000000e+000**	1.781358e-013	1.243182e-009	1.920520e+001	8.228709e-003	5.459585e+001
	Best	0.000000e+000	0.000000e+000	1.136870e-013	2.832990e+001	3.709010e-003	1.944010e+002
	Worst	0.000000e+000	9.663380e-013	5.659500e-009	1.167770e+002	3.569160e-002	4.253200e+002
	Rank	1	2	3	4	5	6
f_8	Mean	6.711509e+002	1.281399e+003	**2.338847e+001**	6.066194e+003	2.962787e+002	5.488050e+003
	Std	7.550727e+002	8.304642e+002	**4.426158e+001**	9.074134e+002	1.637976e+002	9.252262e+002
	Best	3.818270e-004	3.749340e-001	3.818280e-004	4.765680e+003	1.780470e-003	3.394480e+003
	Worst	2.842520e+003	2.986510e+003	1.184660e+002	8.114360e+003	5.921940e+002	7.212190e+003
	Rank	3	4	1	6	2	5
f_9	Mean	**6.335673e-015**	6.809368e-015	6.412648e-014	2.030707e+000	1.547524e-002	2.015904e+001
	Std	**1.802705e-015**	1.703396e-015	9.207796e-015	5.647839e-001	3.958397e-003	2.491498e-001
	Best	4.440892e-015	4.440890e-015	4.352070e-014	9.468440e-001	9.894610e-003	1.955740e+001
	Worst	7.993606e-015	7.993610e-015	8.260060e-014	3.099410e+000	3.233120e-002	2.065720e+001
	Rank	1	2	3	5	4	6
f_{10}	Mean	**8.216455e-004**	5.442769e-003	9.636020e-004	1.959137e-001	2.982075e-002	6.558661e+000
	Std	**2.534696e-003**	1.663440e-002	2.870144e-003	7.146373e-002	3.222232e-002	1.804541e+000
	Best	0.000000e+000	0.000000e+000	1.110220e-016	8.263810e-002	1.484270e-003	4.095480e+000
	Worst	9.857285e-003	5.859020e-002	1.232100e-002	3.517030e-001	1.519410e-001	1.118610e+001
	Rank	1	3	2	5	4	6
f_{11}	Mean	1.010637e+002	**8.292695e+001**	1.093846e+002	1.217936e+002	1.518107e+002	3.203635e+002
	Std	1.758895e+001	**5.423910e+001**	1.484148e+001	3.401371e+001	3.681696e+001	6.293478e+001
	Best	7.011769e+001	3.665610e+001	7.574960e+001	5.809480e+001	8.059630e+001	1.918650e+002
	Worst	1.471888e+002	1.932100e+002	1.393060e+002	1.786960e+002	2.388030e+002	5.088410e+002
	Rank	2	1	3	4	5	6
f_{12}	Mean	**2.448002e+003**	2.518676e+003	3.325394e+003	6.135356e+003	4.792848e+003	6.470636e+003
	Std	**3.716220e+002**	4.312037e+002	1.979133e+002	8.832789e+002	6.660686e+002	5.115047e+002
	Best	1.812773e+003	1.739860e+003	2.958040e+003	4.305520e+003	3.436130e+003	5.371710e+003
	Worst	3.163769e+003	3.413160e+003	3.806870e+003	7.779220e+003	6.534880e+003	7.437780e+003
	Rank	1	2	**3**	5	4	6
f_{13}	Mean	**6.098825e-015**	6.690944e-015	2.474520e+000	2.226891e+000	1.006353e+001	1.990731e+001
	Std	**2.420980e-015**	1.741301e-015	5.849789e-001	7.625523e-001	8.596402e+000	4.249608e-001
	Best	4.440892e-015	4.440890e-015	1.175130e+000	6.470250e-001	1.671650e-002	1.890770e+001
	Worst	1.509903e-014	7.993610e-015	3.763000e+000	3.646700e+000	1.952490e+001	2.056130e+001
	Rank	1	2	4	3	5	6
f_{14}	Mean	**5.551115e-017**	1.016895e-002	6.899128e-009	1.927589e-001	6.662652e-002	6.567002e+000
	Std	**2.636150e-016**	2.419241e-002	3.778342e-008	7.129116e-002	6.511676e-002	2.075111e+000
	Best	0.000000e+000	0.000000e+000	5.551120e-016	6.650990e-002	5.309290e-003	3.330400e+000
	Worst	1.443290e-015	1.019360e-001	2.069490e-007	3.563320e-001	2.786410e-001	1.239520e+001
	Rank	1	3	2	5	4	6

Table 6 (*Continued*)

Function		ABO1	ABO2	ABC	PSO	CPSO	GA
f_{15}	Mean	9.747747e+002	1.390672e+003	**9.421086e+002**	1.023783e+003	1.037929e+003	2.430684e+003
	Std	8.073741e+001	3.293194e+002	**5.206952e+001**	7.943480e+001	9.878428e+001	3.640379e+002
	Best	9.000000e+002	1.077860e+003	8.151550e+002	8.076960e+002	8.011960e+002	1.723360e+003
	Worst	1.143544e+003	2.507540e+003	1.000000e+003	1.143560e+003	1.143550e+003	3.073530e+003
	Rank	2	5	1	3	4	6
f_{16}	Mean	**9.058588e+002**	3.161484e+003	9.153301e+002	6.108339e+003	9.688962e+002	7.051675e+003
	Std	**3.624031e+001**	5.070232e+002	3.668016e+001	8.955133e+002	8.093848e+001	8.567972e+002
	Best	8.226412e+002	2.207190e+003	8.143760e+002	4.583140e+003	8.031290e+002	5.484880e+003
	Worst	9.504475e+002	4.192700e+003	1.039040e+003	8.324640e+003	1.145300e+003	8.919990e+003
	Rank	1	4	2	5	3	6
f_{17}	Mean	**6.050445e+003**	8.323339e+003	6.483812e+003	6.724918e+003	7.641499e+003	9.367648e+003
	Std	**5.493975e+002**	4.874880e+002	3.917772e+002	9.484499e+002	1.011206e+003	3.492470e+002
	Best	5.152314e+003	7.052840e+003	5.483090e+003	4.296450e+003	5.590070e+003	8.723980e+003
	Worst	7.460408e+003	9.136400e+003	7.117980e+003	8.281350e+003	9.648700e+003	1.001430e+004
	Rank	1	**5**	2	3	4	6
f_{18}	Mean	**1.290269e+003**	1.293506e+003	1.295599e+003	1.313120e+003	1.308917e+003	1.329262e+003
	Std	**4.918307e+000**	1.398559e+001	5.949596e+000	2.607855e+001	9.593911e+000	5.246180e+000
	Best	1.279854e+003	1.259090e+003	1.283760e+003	1.273840e+003	1.287080e+003	1.316900e+003
	Worst	1.300767e+003	1.304970e+003	1.305460e+003	1.418680e+003	1.326050e+003	1.337180e+003
	Rank	1	2	3	5	4	6
f_{19}	Mean	1.3959360e+003	**1.395840e+003**	1.410295e+003	1.441987e+003	1.432338e+003	1.464964e+003
	Std	6.476220e+000	**1.373798e+001**	5.389545e+000	2.514585e+001	1.808164e+001	9.922366e+000
	Best	1.383456e+003	1.368210e+003	1.397020e+003	1.406960e+003	1.400800e+003	1.441310e+003
	Worst	1.406612e+003	1.412910e+003	1.419530e+003	1.534240e+003	1.474890e+003	1.483010e+003
	Rank	2	1	3	5	4	6
f_{20}	Mean	1.401264e+003	1.491011e+003	**1.401158e+003**	1.536440e+003	1.577127e+003	1.535247e+003
	Std	5.792433e-001	9.632108e+001	**3.387940e-001**	8.412452e+001	6.062380e+001	6.974717e+001
	Best	1.400632e+003	1.400450e+003	1.400470e+003	1.400070e+003	1.400380e+003	1.420430e+003
	Worst	1.402942e+003	1.603520e+003	1.401870e+003	1.621400e+003	1.612650e+003	1.611660e+003
	Rank	2	3	1	5	6	4
f_{21}	Mean	2.445707e+003	2.550871e+003	**1.728651e+003**	2.542164e+003	2.675666e+003	2.704342e+003
	Std	5.158269e+001	8.268553e+001	**1.492234e+002**	1.226510e+002	9.547055e+001	3.517500e+001
	Best	2.304198e+003	2.281060e+003	1.700000e+003	2.277310e+003	2.477110e+003	2.634280e+003
	Worst	2.525535e+003	2.632950e+003	2.518620e+003	2.809930e+003	2.863420e+003	2.760090e+003
	Rank	2	4	1	3	5	6
f_{22}	Mean	**1.700000e+003**	2.280637e+003	1.739832e+003	4.427291e+003	3.504812e+003	4.159887e+003
	Std	**8.433730e-004**	1.988165e+002	2.187440e+002	1.116674e+003	1.628007e+003	2.774419e+002
	Best	1.700000e+003	2.035590e+003	1.571040e+003	1.713280e+003	1.501750e+003	3.614040e+003
	Worst	1.700005e+003	2.934750e+003	2.823650e+003	5.885220e+003	6.930810e+003	4.927150e+003
	Rank	1	3	2	6	4	5

than ABO1 in terms of accuracy. ABO1, CPSO, PSO and ABC converged fast at first, and then converged very slowly.

On Rotated Schwefel function, the performance order for the six intelligent algorithms is ABO1>ABO2>ABC>CPSO>PSO>GA. On Rotated Schwefel function, the convergence speed of ABO1 and ABO2 is better than the other algorithms.

On Rotated Ackley function, the performance order for the six intelligent algorithms is ABO1>ABO2>PSO>ABC>CPSO>GA. On Rotated Ackley function, ABO1 and ABO2 are 15 orders of magnitude better than PSO and ABC in terms of accuracy. ABO1 and ABO2 are 14 orders of magnitude better than CPSO and GA in terms of accuracy.

On Rotated Griewank function, the performance order for the six intelligent algorithms is ABO1>ABC>ABO2>CPSO>PSO>GA. PSO, CPSO, ABO2 and GA perform poorly with the Rotated Griewank function. ABC and ABO1 are similar with each other before about 80,000FEs. After about 80,000FEs, the convergence speed of ABO1 performs from slow to fast. The result of ABO1 improved continually and obtained the best results, seen in Fig. 6 (f_{14}).

As seen in Fig. 6 (f_{13}) and (f_{14}), PSO, CPSO and GA are sensitive to the Rotated Ackley function and the Rotated Griewank function, and these three algorithms hardly converged.

3) Composition functions

On eight composition functions, all algorithms converged fast at first, and then converged very slowly. However, ABO1 got the best results on four functions and got the second place on four other functions.

From the above analysis, we can see ABO1 algorithm and ABO2 algorithm outperform the other three algorithms in terms of accuracy, robustness, and convergence speed on most test functions. On four rotated functions, performances of ABO1 and ABO2 are much similar with those on non-rotated ones. So it shows ABO1 and ABO2 are not sensitive to rotation and are able to maintain their excellent performance abilities on these four rotated functions.

Table 7
Results of the Iman-Davenport test.

Dimension	Iman-Davenport	Critical value $\alpha = 0.05$	Significant differences?
30	44.0109	2.29∼2.37	Yes

4.2.4. Statistical analysis

It is obvious that ABO1 got the best ranking with dimension of 30. Statistical evaluation of experimental results has been considered an essential part of validation of new intelligent methods. The Iman-Daveport and Holm tests are non-parametric statistical methods and used to analyze the behaviors of evolutionary algorithms in many recent works. The Iman-Daveport and Holm tests are used in this section. Details of the two statistical methods are introduced in reference [28]. The results of the Iman-Davenport test are showed in Table 7. The values are distributed according to F-distribution with 5 and 105° of freedom. The critical values are looked up in the F-distribution table with level of 0.05. As can be seen in Table 7, the Iman-Davenport values are larger than their critical values which mean that significant differences exist among the rankings of the algorithms.

Holm test was employed as a post hoc procedure. The ABO1 algorithm was chosen as the control algorithm. The results of Holm

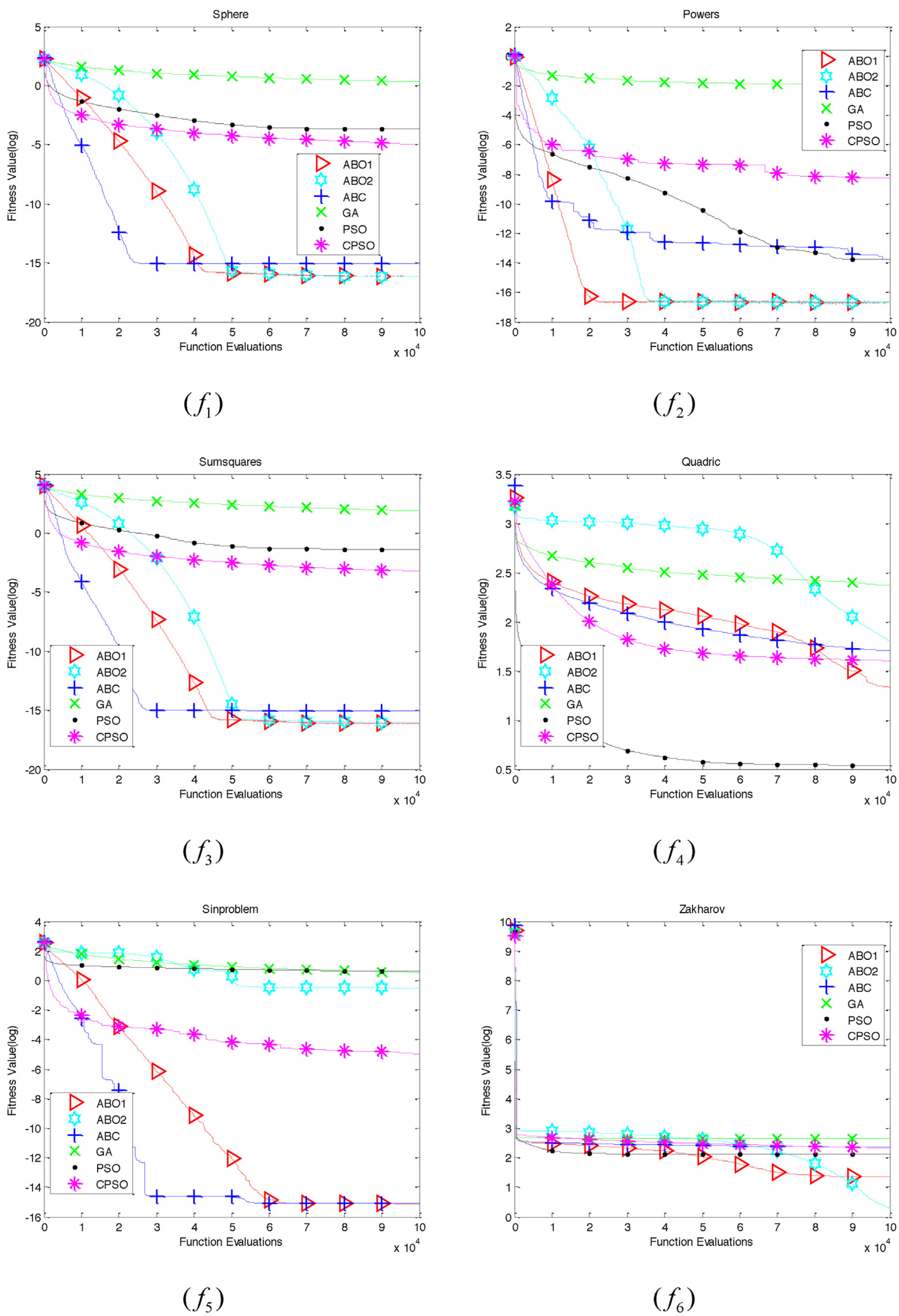


Fig. 6. The mean best function value profiles of ABO1, ABO2, ABC, GA, PSO and CPSO.

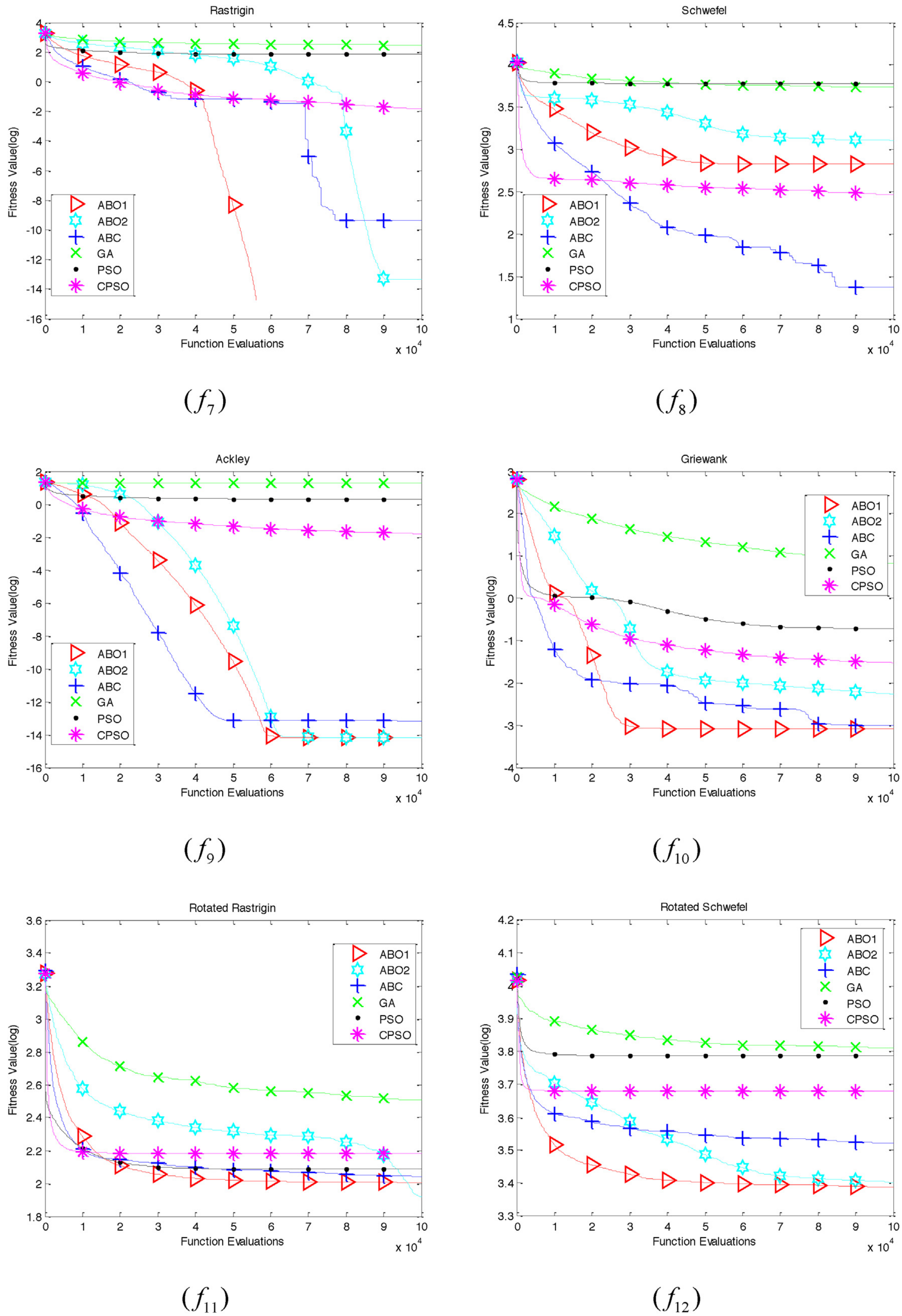


Fig. 6. (*Continued*)

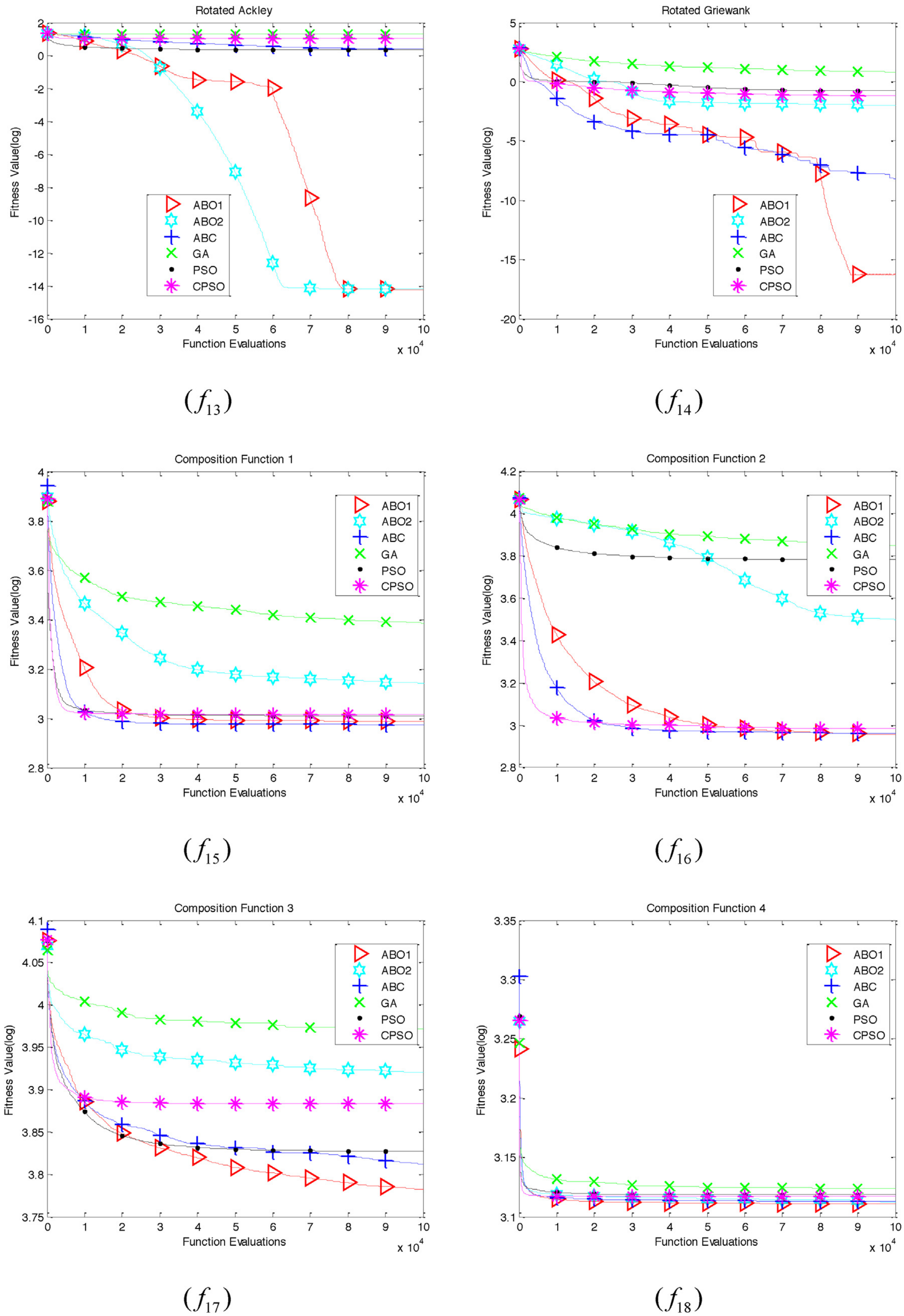


Fig. 6. (*Continued*)

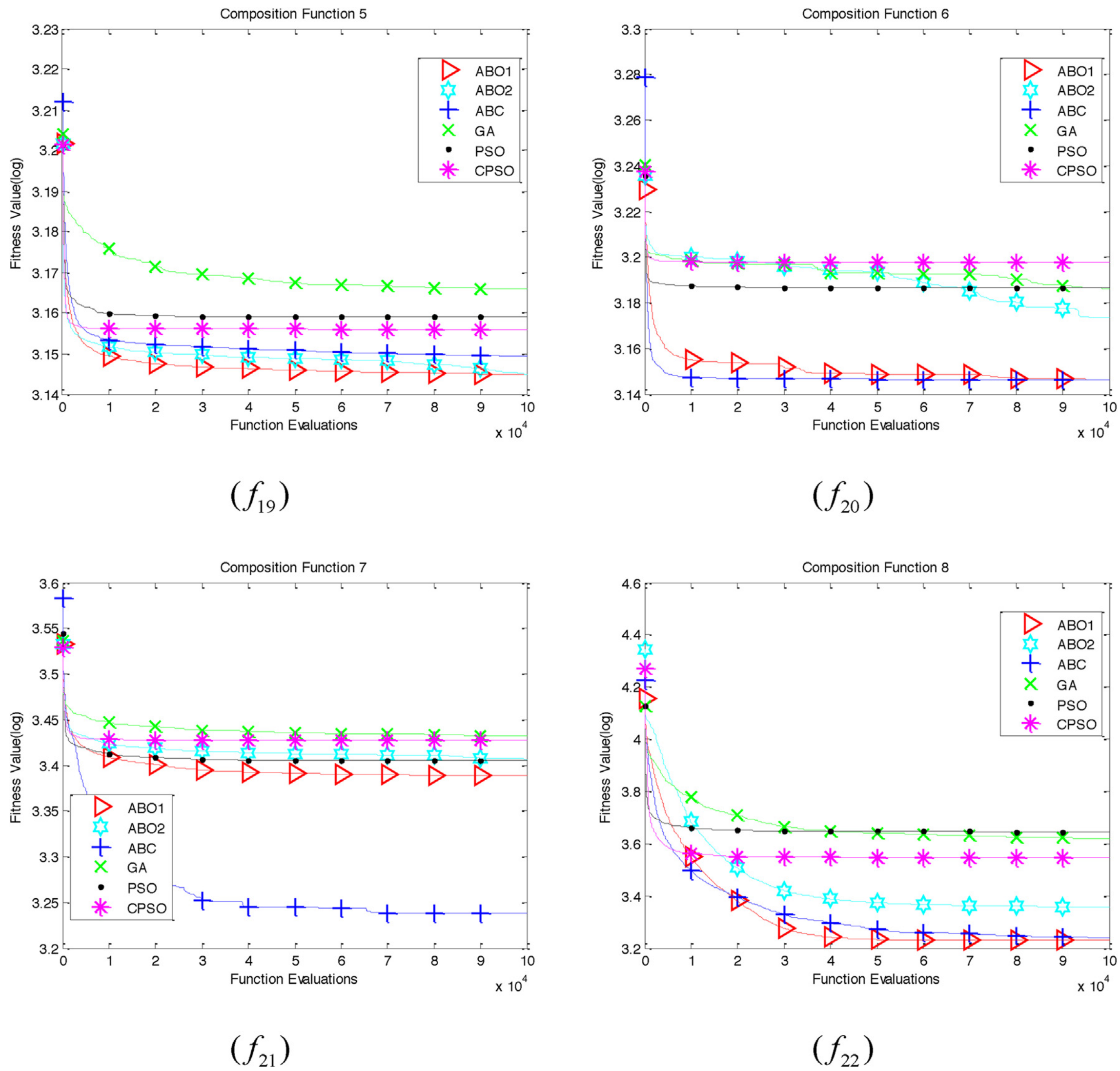


Fig. 6. (*Continued*)

Table 8
Comparison (Holm's test) of ABO1 with the remaining algorithms.

Algorithm	z	p value	α/i	Significant differences?
GA	7.7359	1.0267E-14	0.01	Yes
PSO	5.1572	2.5058E-7	0.0125	Yes
CPSO	4.7543	1.9908E-6	0.0167	Yes
ABO2	2.4980	0.0124	0.025	Yes
ABC	2.0951	0.0361	0.05	Yes

tests are given in Table 8. The α/i values listed in the tables are with level of 0.05.

ABO1 got the best ranking and is the control algorithm. As seen in Table 8, the p values of GA, PSO, CPSO, ABO2 and ABC are smaller than their α/ivalues, which means that equality hypotheses are rejected and significant differences exist between these five algorithms and the control algorithm.

4.2.5. Algorithm complexity analysis

In many heuristic algorithms, most of the computation is spent on fitness evaluation in each generation. Assuming that the computation cost of one individual in ABO is *Cost_a* and the cost of the sorting is *Cost_s*, *n* is the population size, then, the total computation cost of ABO for one generation is $n{\cdot}Cost\_a + Cost\_s$. *Cost_a* is associated with the test function complexity. It is very difficult to give a brief analysis in terms of time for all algorithms. Through directly evaluating the algorithmic time response on different benchmark functions (f_1-f_{10}), the average computing time in 30 sample runs of all algorithms is given in Fig. 7. From the results, it is observed that CPSO takes the most computing time in all compared algorithms. PSO takes the least computing time in all compared algorithms. ABO1, ABO2 and ABC are about the same time. In summary, it is concluded that, compared with other algorithms, the ABO1 algorithm requires less computing time to achieve better results.

5. Conclusion

Artificial Butterfly Optimization (ABO) algorithm, based on the mate-finding behavior of speckled woods, is presented in this paper. The overall efficiency of a bio-inspired algorithm depends on a good balance between exploration and exploitation. In the ABO algorithm, all virtual butterflies are divided into two groups, one called sunspot butterfly group and the other called canopy butterfly group. The fitness of sunspot butterflies is better than those of canopy butterflies. In an optimization process, exploration is achieved by canopy butterflies while exploitation is achieved by sunspot butterflies. Furthermore, the flight strategies for sunspot

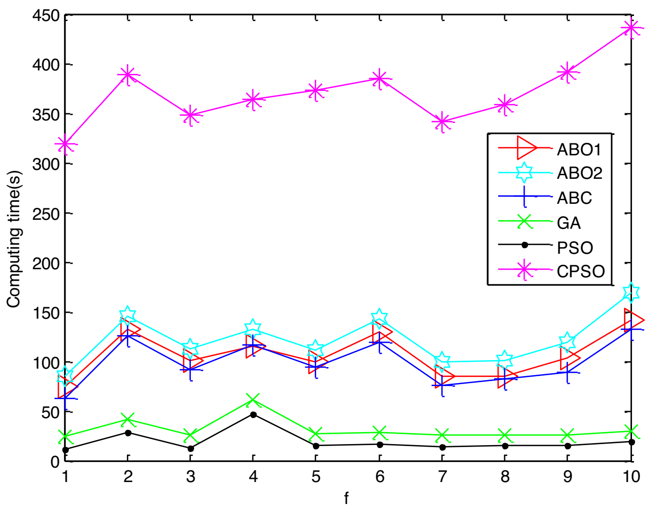


Fig. 7. Computing time of all algorithms on different problems.

butterflies and canopy butterflies can be redefined. Different flight strategies lead to different algorithm instances. From this point, ABO is a mimic-life algorithm in grandness.

By setting different flight strategies in the ABO algorithm, two new algorithms named ABO1 algorithm and ABO2 algorithm are built. Twenty-two benchmark functions with dimensions of 30 were used to compare with four well-known metaheuristic algorithms including ABC, PSO, CPSO and GA. The numerical experimental results show the performance of ABO1 was superior to ABC, PSO, CPSO and GA on most benchmark functions. Iman-Davenport and Holm tests were used for statistic analysis, it also indicate that ABO1 is significant better than the rival algorithms. In other words, artificial sunspot butterflies and artificial canopy butterflies can strike a balance between the exploration and the exploitation of the search space just like real butterflies. It also proves that ABO is potentially more powerful as an optimizing algorithm framework.

A further extension to the current ABO algorithm may lead to even more effective optimization algorithms. Therefore, future research efforts will be focused on finding new flight strategies to build more effective algorithms based on ABO and applying the proposed algorithm to solve practical engineering problems.

Acknowledgments

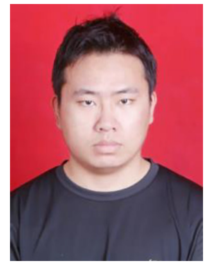
This work is supported by the National Natural Science Foundation of China (Grantno.61174164, 51205389) and the Natural Science Foundation of Liaoning Province(2015020163). And the authors are very grateful to the anonymous reviewers for their valuable suggestions and comments to improve the quality of this paper.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of IEEE Int Conference on Neural Network (1995) 1942–1948.
[2] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, Frome, UK, 2010.
[3] X. Li, Z. Shao, J. Qian, An optimizing method based on autonomous animats: fish-swarm algorithm, Syst. Eng. Theory Practice 22 (2002) 32–38.
[4] D. Karaboga, An idea based on honey bees warm for numerical optimization, technical report-TR06, Erciyes University, Engineering Faculty Comput. Eng. Dep., 2005.
[5] J.H. Holland, Adaptation in Natural and Artificial Systems Ann Arbor, University of Michigan Press, 1975.
[6] K.M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, IEEE Control Syst. Mag. 22 (2002) 52–67.
[7] W. Cai, W. Yang, X. Chen, A global optimization algorithm based on plant growth theory: plant growth optimization, International Conference on Intelligent Computation Technology and Automation (ICICTA) (2008) 1194–1199.
[8] D.H. Wolpert, The supervised learning no-free-lunch theorems, the Sixth Online World Conference on Soft Computing in Industrial Applications (2001).
[9] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67–82.
[10] Darrell J. Kemp, C. Wiklund, H.V. Dyck, Contest behaviour in the speckled wood butterfly (Pararge aegeria): seasonal phenotypic plasticity and the functional significance of flight performance, Behav. Ecol. Sociobiol. (2006) (403-401).
[11] M.J. Goddard, Broods of the speckled wood (Pararge aegeria aegerides stgr.) lepidoptera: satyridae), Entomologist (1962) 289–307.
[12] M.J. Goddard, Broods of the speckled wood (Pararge aegeria aegerides Stgr.) (Lepidoptera: satyridae), Entomologist (1967) 241–254.
[13] L.V. Velde, C. Turlure, H. Van Dyck, Body temperature and territory selection by males of the speckled wood butterfly (Pararge aegeria): what makes a forest sunlit patch a rendezvous site, Ecol. Ent. 36 (2011) 161–169.
[14] M. Bergman, C. Wiklund, Differences in mate location behaviours between residents and nonresidents in a territorial butterfly, Anim. Behav. 78 (2009) 1161–1167.
[15] N.B. Davies, Territorial defence in the speckled wood butterfley (PARARGE AEGERIA): the resident always wins, Anim. Behav. (1978) 138–147.
[16] D. Kemp, C. Wiklund, K. Gotthard, Life history effects upon contest behaviour: age as a predictor of territorial contest dynamics in two populations of the speckled wood butterfly, Pararge aegeria L, Ethology 112 (2006) 471–477.
[17] E.L. Yu, P.N. Suganthan, Ensemble of niching algorithms, Inform. Sci. 180 (15) (2010) 2815–2833.
[18] M.Z. Ali, N.H. Awad, A novel class of niche hybrid cultural algorithms for continuous engineering optimization, Inform. Sci. 267 (2014) 158–190.
[19] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Software 95 (2016) 51–67.
[20] M. Clerc, J. Kennedy, The particle swarm-explosion, stability, and convergence in a multidimensional complex space, IEEE Trans. Evol. Comput. 6 (no. 1) (2002) 58–73.
[21] F. van den Bergh, A.P. Engelbrecht, A cooperative approach to participle swam optimization, IEEE Trans. Evol. Comput. 8 (no. 3) (2004) 225–239.
[22] S. Sumathi, T. Hamsapriya, P. Surekha, Evolutionary Intelligence An Introduction to Theory and Applications with Matlab, Springer-Verlag, Berlin, 2008.
[23] B. Niu, Y.L. Zhu, X.X. He, H. Wu, MCPSO. A multi-swarm cooperative particle swarm optimizer, Appl. Math. Comput. 185 (2) (2007) 1050–1062.
[24] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. 214 (2009) 108–132.
[25] J. J. Liang, B-Y. Qu, P. N. Suganthan, Alfredo G. Hernández-Díaz, Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization, Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University Singapore, J anuary 2013.
[26] J.J. Liang, A.K. Qin, P.N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, IEEE Trans. Evol. Comput. 10 (3) (2006) 281–295.
[27] R. Salomon, Reevaluating genetic algorithm performance under coordinate rotation of benchmark functions, Bio Syst. 39 (1996) 263–278.
[28] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inform. Sci. 180 (2010) 2044–2064.

Xiangbo Qi received his M.S. degree from Chinese Academy of Sciences, China in 2007. He is currently pursuing the Ph.D. degree at Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. His current research interests include bio-inspired computation, bioinformatics and the application of the intelligent optimization algorithms on data mining and scheduling.

Yunlong Zhu is the Director of Dept. of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences. He received his Ph.D. degree from Chinese Academy of Sciences, China in 2005. He has research interests in various aspects of Enterprise Information Management but he has ongoing interests in artificial intelligence, data mining, complex systems and related areas. Prof. Zhu¡s̄ research has led to a dozen professional publications in these areas.



Hao Zhang received the B.Sc. degree in Mechanical Design and Automation from Dalian University of Technology, Dalian, China, in 2007. He received his Ph.D. degree in mechanical and electronic engineering from Chinese Academy of Sciences, China in 2014. He has been an Assistant Research Fellow in Dept. of Information Service & Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. His current research interests include swarm intelligence, bioinformatics and other stochastic optimization methods.