A new hybrid artificial bee colony algorithm for global optimization

Xiangyu Kong[1], Sanyang Liu[2], Zhen Wang[3]

[1.] Department of Applied Mathematics, Xidian University, Xi'an 710071, China

[2.] Department of Applied Mathematics, Xidian University, Xi'an 710071, China

[2.] Institute of Information and System Computation Science, Beifang University of Nationalities, Yinchuan 750021, China

Abstract

To further improve the performance of artificial bee colony algorithm (ABC), a new hybrid ABC (HABC) for global optimization is proposed via exploring six initialization methods. Furthermore, to balance the exploration and exploitation abilities, a new search mechanism is also developed. The algorithms are applied to 27 benchmark functions with various dimensions to verify its performance. Numerical results demonstrate that the proposed algorithms outperforms the ABC in global optimization problems, especially the HABC algorithm with random initialization and HABCO algorithm with orthogonal initialization.

Keywords: *Artificial bee colony algorithm, Initialization methods, Search mechanism, Differential evolution.*

1. Introduction

Global optimization problems arise in almost every field of science, engineering and business. By now, learning from life system, many optimization methods have been developed to solve global optimization problems, such as genetic algorithms (GAs) [1,2], ant colony optimization (ACO) [3], differential evolution (DE) [4] and particle swarm optimization (PSO)[5]. These kinds of algorithms can be named as artificial-life computation. Recently, Karaboga [6] proposed a new kind of optimization technique called artificial bee colony (ABC) algorithm for global numerical function optimization, which simulates the foraging behavior of honey bee swarm. A set of comparison experimental results show that ABC algorithm is competitive to some conventional bio-inspired algorithms with an advantage of employing fewer control parameters [7]. Due to its simplicity, ABC algorithm has been applied to solve many kind of real-world problems, for instance, leaf-constrained minimum spanning tree problem [8], flow shop scheduling problem [9], inverse analysis problem [10], radial distribution system network reconfiguration problem [11], clustering problem [13], TSP problems [14], and so on.

According to the applications showed above, ABC algorithm seems to be a well-performed algorithm. However, similar to other population-based algorithms, there still are insufficiencies in ABC algorithm, such as slower convergence speed for some unimodal problems and easily get trapped in local optima for some complex multimodal problems [7]. It is well known that for the population-based algorithms the exploration and the exploitation abilities are both necessary facts. The exploration ability refers to the ability to investigate the various unknown regions to discover the global optimum in solution space, while the exploitation ability refers to the ability to apply the knowledge of the previous good solutions to find better solutions. The exploration ability and the exploitation ability contradict to each other, so that the two abilities should be well balanced to achieve good performance on optimization problems. So far as we know that the search equation of ABC algorithm is good at exploration but poor in exploitation.

Therefore, accelerating convergence speed and avoiding local optima have become two most important goals in ABC algorithm modification. To overcome the issues in ABC algorithm and achieve the two goals above, inspired by PSO and DE, a new search mechanism is proposed in the new hybrid artificial bee colony (HABC) algorithm. In order to balance the exploration ability and the exploitation ability, the random search equation is used in the employed bee stage and the best-guided search equation is used in the onlooker bee stage. In addition, to enhance the convergence speed, six initialization methods are employed and compared, including random initialization, chaotic initialization, opposition-based initialization, inter-cell initialization, chaotic opposition-based initialization, and orthogonal initialization [15]. Experimental results and comparisons denote the effectiveness and efficiency of the proposed HABC algorithms.

The rest of the paper is organized as follows. In Section 2, ABC algorithm is summarized briefly. In Section 3, the proposed hybrid artificial bee colony algorithm is

described. In Section 4, experiments are presented and the results are discussed. Finally, a conclusion is provided in Section 5.

2. Overview of artificial bee colony algorithm

By simulating the foraging behavior of bee colonies, artificial bee colony (ABC) algorithm, which is a swarm intelligence-based optimization algorithm, was proposed by Karaboga in 2005 for numerical function optimization [6]. The main steps of ABC algorithm can be described as follows.

Initialization
Repeat
  Employed bee stage: Place the employed bees on the food sources in the memory.
  Onlooker bee stage: Place the onlooker bees on the food sources in the memory.
  Scout bee stage: Send the scout bees to the search area for discovering new food sources.
Until (conditions are satisfied)

In ABC algorithm, the colony consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony is employed bees, and the other half is onlooker bees. The employed bees explore the food source and send the information of the food source to the onlooker bees. The onlooker bees choose a food source to exploit based on the information shared by the employed bees. The scout bee, which is one of the employed bees whose food source are abandoned, finds a new food source randomly. The position of a food source is a possible solution to the optimization problem. Denote the food source number as SN, the position of the i th food source as x_i $(i=1,\cdots,SN)$, which is a D-dimensional vector.

In ABC algorithm, the i th fitness value fit_i for a minimization problem is defined as:

$$fit_i = \begin{cases} \dfrac{1}{1+f_i}, & f_i \geq 0, \\ 1+abs(f_i), & f_i < 0, \end{cases} \tag{2.1}$$

where f_i is the cost value of the i th solution.

The probability of a food source being selected by an onlooker bee is given by:

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i}. \tag{2.2}$$

A candidate solution from the old one can be generated as:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \tag{2.3}$$

where $k \in \{1,2,\cdots,SN\}$, $k \neq i$ and $j \in \{1,2,\cdots,D\}$ are randomly selected indices, $\phi_{ij} \in [-1,1]$ is a uniformly distributed random number. The candidate solution is compared with the old one, and the better one should be remained.

If the abandoned food source is x_i, the scout bee exploits a new food source according to:

$$x_{ij} = x_j^{\min} + rand(0,1)\left(x_j^{\max} - x_j^{\min}\right), \tag{2.4}$$

where $x_j^{\max}$ and $x_j^{\min}$ are the upper and lower bounds of the j th dimension of the problem's search space.

3. New Hybrid artificial bee colony algorithm

3.1 Initialization

Population initialization is a crucial step in swarm intelligence algorithms, because it can affect the quality of the solutions and the convergence speed. There are several kinds of initialization methods to generate the initial population. Here, the following initialization methods will be considered in the new hybrid artificial bee colony algorithm, and the efficiency of these methods will be compared.

3.1.1 Random initialization

The random initialization is the most commonly used method to generate initial population. In the original ABC algorithm, the random initialization was employed. The initial population is generated randomly within the range of the boundaries of the parameters, which is:

$$x_{ij} = x_j^{\min} + rand(0,1)\left(x_j^{\max} - x_j^{\min}\right), \tag{3.1}$$

where $i \in \{1,2,\cdots,SN\}$, $j \in \{1,2,\cdots,D\}$.

3.1.2 Chaotic initialization

Chaos is found in non-linear dynamical systems, which is a deterministic random-like process. Chaos is apparently random and unpredictable but it also presents regularity, and it has a very sensitive dependence upon the initial condition and parameters. Mathematically, chaotic maps may be considered as source of randomness. Because of these randomness and sensitivity dependence on the initial conditions of chaotic maps, it has been considered as an initialization method for the heuristic algorithms to improve the global convergence by escaping from local optima. [16]

A chaotic map is a discrete time dynamic system running in chaotic state, which is: $x_{k+1} = f(x_k)$, $0 < x_k < 1$, $k = 1, 2, \cdots$.

According to the definition of $f(\cdot)$, the chaotic maps include several types, such as logistic map, circle map, gauss map , sinusoidal iterator, and so on. Here, the sinusoidal iterator is employed as a chaotic map in colony initialization. The initial equation is defined as follows:

$$x_{ij} = x_j^{\min} + ch_{k,j}\left(x_j^{\max} - x_j^{\min}\right), \quad (3.2)$$

where $ch_{k,j} = \sin(\pi \cdot ch_{k-1,j})$, $ch_{k-1,j} \in (0,1)$, $k = 1, 2, \cdots, K$, is the chaotic sequence.

3.1.3 Opposition-based initialization

According to [17], random initial solutions are further from the solution than their opposite random initial solutions, which can accelerate convergence. The initial population is generated as follows:

$$ox_{ij} = x_j^{\min} + x_j^{\max} - x_{ij}, \quad (3.3)$$

where $x_{ij} = x_j^{\min} + rand(0,1)\left(x_j^{\max} - x_j^{\min}\right)$.

3.1.4 Chaotic opposition-based initialization

Based on the properties of chaotic maps and opposition-based learning methods, a new initialization approach employs both chaotic maps and opposition-based learning methods [18]. The new initialization approach is given as follows:

$$ox_{ij} = x_j^{\min} + x_j^{\max} - x_{ij}, \quad (3.4)$$

where $x_{ij} = x_j^{\min} + ch_{k,j}\left(x_j^{\max} - x_j^{\min}\right)$, $ch_{k,j} = \sin(\pi \cdot ch_{k-1,j})$, and $ch_{k-1,j} \in (0,1)$, $k = 1, 2, \cdots, K$, is the chaotic sequence.

3.1.5 Inter-cell initialization

In the population initialization step, it is desired that the initial population can be scattered uniformly over the feasible solution space, so that the algorithm can search the whole solution space evenly. For this, the inter-cell initialization can be employed: the feasible solution space is divided into *SN* subspace and a solution is randomly selected in each one of these subspaces. Here, *SN* is the population size.

3.1.6 Orthogonal initialization

An orthogonal array specifies a small number of combinations that are scattered uniformly over the space of all possible combinations. Thus, the initial population generated by the orthogonal design can be scatted uniformly over the feasible solution space, so that the algorithm can explore the solution space evenly. Here, we employ the orthogonal initialization method based on orthogonal array and quantization technique, which is described in [19] [20].

3.2 New search mechanism

It is well known that both the exploration and exploitation abilities are necessary for the population based algorithms. How to balance these two abilities to achieve good optimization performance is very important.

In section 2, in ABC algorithm, the employed bees explore the new food source and send the information to the onlooker bees; while the onlooker bees exploit the food sources which are explored by the employed bees. It means that in the ABC algorithm the employed bee stage represents the exploration ability of the algorithm, and the onlooker bee stage represents the exploitation ability of the algorithm. The search equation proposed in ABC algorithm is good at exploration but poor at exploitation, so that it will affect the convergence speed of the algorithm. Inspired of PSO [5], in order to improve the exploitation ability of ABC algorithm, take the advantages of the search equation in PSO, the global best solution will be considered in the new search equation in the onlooker bee stage. The modified search equation in onlooker bee stage is described as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(y_j - x_{ij}), \quad (3.5)$$

where $k \in \{1, 2, ..., SN\}$ is a random selected index which is different from i , $j \in \{1, 2, ..., D\}$ is a random selected index, y_j is the j th element of the global best solution, $\phi_{ij} \in [-1, 1]$ and $\varphi_{ij} \in [0, 1.5]$ are both uniformly distributed random numbers.

Differential evolution (DE) [4] is a population based algorithm to function optimization, whose main strategy is to generate a new position for an individual by calculating vector differences between other randomly selected members in the population. “DE/current-to-rand/1” is a variant DE mutation strategy, which can effectively maintain population diversity according to randomness of the search equation. Motivated by “DE/current-to-rand/1” mutation strategy and based on the property of ABC algorithm, a new search equation in employed bee stage is proposed as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_{ij} - x_{lj}), \quad (3.6)$$

where $k, l \in \{1, 2, ..., SN\}$ are random selected indexes which are different from i ; $j \in \{1, 2, ..., D\}$ is a random selected index; $\phi_{ij} \in [-1, 1]$ and $\varphi_{ij} \in [-1, 1]$ are uniformly

distributed random number; ϕ_{ij} and φ_{ij} are both negative or both positive, which can keep the search direction the same.

In general, inspired by DE and PSO, the new search equation and search mechanism are proposed to balance the exploration ability and exploitation ability in ABC algorithm. In the employed bee stage, search equation (3.6) is used to keep the exploration ability of ABC algorithm; while, in the onlooker bee stage, search equation (3.5) is employed to increase the exploitation ability of the algorithm.

3.3 The hybrid artificial bee colony algorithm

Based on the above analysis, the main steps of the new hybrid artificial bee colony are as follows.

Algorithm: Hybrid artificial bee colony algorithm

Initialize the food sources by using one of the initialization methods proposed in subsection 3.1, and evaluate the population, $trail_i = 0$, $(i = 1, 2, \cdots, SN)$. $Cycle = 1$.

Repeat

Step 1: Search the new food source for employed bee according to (3.6) and evaluate its quality.

Step 2: Apply a greedy selection process and select the better solution between the new food source and the old one.

Step 3: If solution does not improve $trail_i = trail_i + 1$, otherwise $trail_i = 0$.

Step 4: Calculate the probability according to (2.2) and apply roulette wheel selection scheme to choose a food source for onlooker bees.

Step 5: Search the new food source for onlooker bees according to (3.5) and evaluate its quality.

Step 6: Apply a greedy selection process and select the better solution between the new food source and the old one.

Step 7: If solution does not improve $trail_i = trail_i + 1$, otherwise $trail_i = 0$.

Step 8: If $\max(trail_i) > \text{limit}$, replace this food source with a new food source produced by the initialization methods proposed in subsection 3.1.

Memorize the best solution achieved so far.

$Cycle = Cycle + 1$

Until($Cycle =$ Maximum Cycle Number)

4. Numerical experiments

4.1 Test functions and parameter settings

In this section, the HABC algorithm with six different initialization methods is applied to minimize 27 benchmark functions, as shown in Table 1, 2 and 3. In Table 1 and 2, the dimensions of the benchmark functions are given in the third column. The benchmark functions presented in Table 3 are tested of dimension $D = 30$ and dimension $D = 100$. All the benchmark functions, presented in Table 1, 2 and 3, include unimodal, multimodal, regular, irregular, separable, non-separable and multidimensional. Initial range, characteristics and formulation of these functions are listed in Table 1-3.

In all these benchmark functions, Colville and Rosenbrock have an arrow curving valley. If the direction changes cannot be kept up with and the search space cannot explored properly, these two problems will be hard to optimize. The functions of Schwefel and Zakharov have a high eccentric ellipse. The main difficulty is that their gradients are not oriented along the axes, which needs to balance the orientation and high eccentricity of the ellipse makes it a significant challenge for some algorithms. Ackley, Griewank, Rastrigin and Schwefel are complex multimodal functions with a large number of local optima. To obtain good results for these functions, the search strategy must efficiently balance the exploration and exploitation abilities.

All experiments were repeated 25 times independently for each function. The population size was 100 and the maximum number of generation was 3000 for both ABC algorithm and six HABC algorithms in the experiments. Therefore, all experiments were run for 300,000 function evaluations.

4.2 Experimental results

In this subsection, a set of experiments tested on 27 benchmark functions were performed, which compared the performance of six HABC algorithms with ABC algorithm. The results are shown in Table 4-7 in terms of best, worst, mean, standard deviation and mean time. In these tables, ABC represents the original ABC algorithm; HABCOB represents the HABC algorithm with the Opposition-based initialization; HABCCH represents the HABC algorithm with the Chaotic initialization; HABCIC represents the HABC algorithm with the Inter-cell initialization; HABCCOB represents the HABC algorithm with the Chaotic opposition-based initialization; HABCO represents the HABC algorithm with the Orthogonal

initialization and HABC represents the HABC algorithm with the random initialization. The best results are highlighted in boldface.

Compared with the results, the mean values of these six HABC algorithms are equal or close to the optimal ones and the standard deviations are relatively small. All seven algorithms, including ABC and other six HABC algorithms can find optimal solutions on fuctions f_2, f_4, f_6, $f_7 - f_9$, $f_{14} - f_{16}$, f_{21}, and f_{22}, f_{24} with $D = 30$. In particular, the HABCCOB algorithm has the smallest standard deviation in function f_7 and f_8, which means

Table 1. Benchmark functions f_1 - f_8 used in experiments. D: Dimension, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable.

No	Range	D	C	Function	Formulation
1	[-4.5,4.5]	2	UN	Beale	$f(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$
2	[-100,100]	2	MS	Bohachevsky	$f(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$
3	[-10,10]	2	MS	Booth	$f(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$
4	[-5,10]× [0,15]	2	MS	Branin	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$
5	[-10,10]	4	UN	Colville	$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2)$
6	[-100,100]	2	UN	Easom	$f(x) = -\cos x_1 \cos x_2 \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$
7	[-2,2]	2	MN	GoldStein-Price	$f(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$
8	[0,1]	3	MN	Hartman3	$f(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]$ $c = [1.0, 1.2, 3.0, 3.2]$; $a = \begin{bmatrix} 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3.0 & 10 & 30 \\ 0.1 & 10 & 35 \end{bmatrix}^T$ $p = \begin{bmatrix} 0.3689 & 0.1170 & 0.2673 \\ 0.4699 & 0.4387 & 0.7470 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{bmatrix}^T$

Table 2. Benchmark functions f_9 - f_{16} used in experiments. D: Dimension, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable, $n = D$.

No	Range	D	C	Function	Formulation
9	[-5,5]	2	MN	Six Hump Camel Back	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
10	[-10,10]	2	UN	Matyas	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
11	[- D , D]	2	MN	Perm	$f(x) = \sum_{k=1}^{n}\left[\sum_{i=1}^{2}(i^k + 0.5)((x_i/i)^k - 1)\right]^2$
12	[-4,5]	4	UN	Powell	$f(x) = \sum_{i=1}^{n/k}(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} + 10x_{4i})^2 + (x_{4i-2} + 10x_{4i-1})^4 + 10(x_{4i-3} + 10x_{4i})^4$
13	[0, D]	24	MN	PowerSum	$f(x) = \sum_{k=1}^{n}\left[\sum_{i=1}^{4}(x_i^k) - b_k\right]^2$, $b = [8,18,44,114]$
14	[0,10]	4	MN	Shekel	$f(x) = -\sum_{j=1}^{m}\left[\sum_{i=1}^{4}(x_i - a_{ij})^2 + c_i\right]^{-1}$, $c = \frac{1}{10}[1,2,2,4,4,6,3,7,5,5]^T$, $a = \begin{bmatrix} 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 5.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 5.0 & 1.0 & 2.0 & 3.6 \\ 4.0 & 1.0 & 8.0 & 6.0 & 3.0 & 2.0 & 3.0 & 8.0 & 6.0 & 7.0 \\ 4.0 & 1.0 & 8.0 & 6.0 & 7.0 & 9.0 & 3.0 & 1.0 & 2.0 & 3.6 \end{bmatrix}$
15	[-10,10]	2	MN	Shubert	$f(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right) \cdot \left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)$
16	[-36,36]	6	UN	Trid6	$f(x) = \sum_{i=1}^{n}(x_i - 1)^2 - \sum_{i=2}^{n} x_i x_{i-1}$

the results obtained by HABCCOB algorithm on these functions are the most stable. And for function f_{14}, the results obtained by HABC algorithm has the smallest standard deviation; for function $f_{15} - f_{21}$, the results obtained by HABCCO have the smallest standard deviation. All these results indicate that on the above 12 functions, the six HABC algorithms obtain the better solutions than the original ABC algorithm. Furthermore, except the function f_1, f_{18} with $D = 30$ and $D = 100$, and f_{23} with $D = 100$, the six HABC algorithms perform better than the original ABC algorithm, while on these three functions the superiority of ABC algorithm to HABC algorithms is not very obvious in terms of the mean and standard deviation of the solutions. These results indicate that HABC algorithms can balance between exploration and exploitation well.

Compared the results obtained by these six HABC algorithm, we can find that the HABC algorithm with the random initialization performs the best, the HABC algorithm with the Orthogonal initialization. Secondly, the HABC algorithm with the Inter-cell initialization is the worst. It indicates that the population initialization methods will affect the solution obtained by the algorithms.

In order to show the performance of the six HABC algorithms more clearly, Figure 1, Figure 2 and Figure 3 show the mean best function value of some functions. It is clear that for most functions the HABC algorithms have the better performance than the ABC algorithm. Particularly, the HABC algorithm performs the best, which can convergence to the optimum fast and stable, and HABCO the second, and HABCOB, HABCCH and HABCCOB perform similar the third, the HABCIC the worst.

Figure 4, Figure 5 and Figure 6 show the statistical results of the function values for the test functions shown in Figure 1 - 3. Here, box plots are used to illustrate the distribution of these samples obtained from 25 independent runs. The upper and lower ends of the box are the upper and lower quartiles. The line within the box represents the median, and thin appendages summarize the spread a shape of the distribution. Symbol " + " indicate for outlier and the notches denote a robust estimation of the uncertainty about the medians for box-to-box comparison. From Figure 4-6, we can see that HABC algorithms can obtain the better and

more stable solutions than ABC algorithm does, especially the HABC algorithm, and further verifies the discussion obtained in Table 4-10 and Figure 1-3.

5. Conclusion

In this paper, a new hybrid artificial bee colony algorithm is developed for global optimization problems with six kind of initial methods and new search mechanism. The experimental results tested on 27 benchmark functions show that HABC algorithms are competitive with ABC algorithm, especially the HABC algorithm. The improvement can mainly be attributed to the following reasons. First, the new search mechanism can balance the exploration and exploitation abilities very well, which can both maintain the diversity and improve the convergent speed. Secondly, the initialization methods can affect the quality of the solutions and the convergence speed. Therefore, the HABC algorithms are accuracy and effective for global optimization problems.

It is desirable to further apply HABC algorithms to solving those more complex real-world optimization problems and it will be our further work.

Acknowledgements

This work is supported by National Nature Science Foundation of China (No. 60974082) and Foundation of State Key Lab. of Integrated Services Networks of China.

References

[1] K. S. Tang, K. F. Man, S. Kwong, Q. He. Genetic algorithms and their applications. IEEE Signal Processing Magazine, 1996, 13(6):22-37.

[2] X. Xue, Y. Gu. Global Optimization Based on Hybrid Clonal Selection Genetic Algorithm for Task Scheduling. Journal of Computational Information Systems. 2010, 6(1):253-261.

Table 3. Benchmark functions f_{17} - f_{27} used in experiments. C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable, $n = D$.

No	Range	C	Function	Formulation
17	[-32,32]	MN	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$
18	[-10,10]	UN	Dixon-Price	$f(x) = (x_1 - 1)^2 + \sum_{i=2}^{n} i(2x_i^2 - x_{i-1})^2$
19	[-600,600]	MN	Griewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
20	[-10,10]	MN	Levy	$f(x) = \sin^2(\pi y_1) + \sum_{i=1}^{n-1}\left[(y_i - 1)^2(1 + 10\sin^2(\pi y_i + 1))\right] + (y_n - 1)^2(1 + 10\sin^2(2\pi y_n))$ $y_i = 1 + \frac{x_i - 1}{4}, \quad i = 1, \cdots, n.$
21	[0, π]	MS	Michalewicz	$f(x) = -\sum_{i=1}^{n}\sin(x_i)\left(\sin\left(ix_i^2/\pi\right)\right)^{2m}$, $m = 10$
22	[-5.12,5.12]	MS	Rastrigin	$f(x) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$
23	[-30,30]	UN	Rosenbrock	$f(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
24	[-500,500]	MS	Schwefel	$f(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$
25	[-100,100]	US	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$
26	[-10,10]	US	SumSquares	$f(x) = \sum_{i=1}^{n} ix_i^2$
27	[-5,10]	UN	Zakharov	$f(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{n} 0.5ix_i\right)^4$

Table 4. Best, worst, mean, standard deviation and mean time value obtained by ABC and other six HABC through 25 independent runs on benchmark function f_1 - f_8 .

		Best	Mean	Worst	Std	Mean-Time
f_1	ABC	**9.77e-17**	**1.98e-15**	**1.64e-14**	**3.06e-15**	**16.33508**
	HABCOB	3.50e-09	3.41e-07	1.76e-06	4.16e-07	16.81662
	HABCCH	1.56e-08	5.71e-07	2.10e-06	6.71e-07	16.71493
	HABCIC	2.37e-09	5.20e-07	5.25e-06	1.05e-06	17.12752
	HABCCOB	3.63e-10	3.22e-07	9.12e-07	2.58e-07	16.92151
	HABCO	1.62e-08	5.35e-07	4.44e-06	8.95e-07	17.23723
	HABC	3.62e-09	5.42e-07	2.92e-06	6.76e-07	17.28006
f_2	ABC	**0**	**0**	**0**	**0**	**16.75061**
	HABCOB	**0**	**0**	**0**	**0**	17.01885
	HABCCH	**0**	**0**	**0**	**0**	16.87816
	HABCIC	**0**	**0**	**0**	**0**	17.38567
	HABCCOB	**0**	**0**	**0**	**0**	17.09213
	HABCO	**0**	**0**	**0**	**0**	17.42189
	HABC	**0**	**0**	**0**	**0**	17.4868
f_3	ABC	4.43e-20	1.79e-18	8.26e-18	1.77e-18	**15.83831**
	HABCOB	4.44e-21	4.41e-19	1.76e-18	3.94e-19	16.22402
	HABCCH	4.11e-20	**3.70e-19**	1.48e-18	**3.86e-19**	16.09317
	HABCIC	**4.06e-21**	5.70e-19	2.86e-18	6.89e-19	16.60484
	HABCCOB	2.00e-20	7.34e-19	3.24e-18	8.12e-19	16.34202
	HABCO	7.78e-20	4.92e-19	**1.35e-18**	4.08e-19	16.68255
	HABC	7.62e-21	4.16e-19	1.54e-18	4.15e-19	16.59208
f_4	ABC	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	**16.20202**
	HABCOB	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	16.59222
	HABCCH	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	16.47659
	HABCIC	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	16.91512
	HABCCOB	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	17.45144
	HABCO	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	17.83527
	HABC	**3.98e-01**	**3.98e-01**	**3.98e-01**	**0**	17.04465
f_5	ABC	5.82e-03	5.79e-02	1.04e-01	2.83e-02	**17.44701**
	HABCOB	1.98e-03	2.72e-02	9.37e-02	2.58e-02	18.1546
	HABCCH	3.54e-03	3.25e-02	1.16 e-01	2.69e-02	18.11104
	HABCIC	**1.21e-03**	**2.40e-02**	**7.26e-02**	**1.99e-02**	18.53727
	HABCCOB	1.77e-03	3.39e-02	1.22e-01	2.96e-02	18.19926
	HABCO	1.46e-03	2.38e-02	9.17e-02	2.29e-02	17.94445
	HABC	1.39e-03	2.79e-02	1.17e-01	2.82e-02	17.62544
f_6	ABC	**-1**	**-1**	**-1**	**0**	**16.53898**
	HABCOB	**-1**	**-1**	**-1**	**0**	16.86348
	HABCCH	**-1**	**-1**	**-1**	**0**	16.70109
	HABCIC	**-1**	**-1**	**-1**	**0**	17.37636
	HABCCOB	**-1**	**-1**	**-1**	**0**	16.91044
	HABCO	**-1**	**-1**	**-1**	**0**	17.21757
	HABC	**-1**	**-1**	**-1**	**0**	17.15267
f_7	ABC	**3**	**3**	**3**	1.46e-15	**16.6893**
	HABCOB	**3**	**3**	**3**	1.21e-15	16.96251
	HABCCH	**3**	**3**	**3**	1.31e-15	16.8877
	HABCIC	**3**	**3**	**3**	1.15e-15	17.20357
	HABCCOB	**3**	**3**	**3**	**5.73e-16**	16.95133
	HABCO	**3**	**3**	**3**	6.72e-16	17.3129
	HABC	**3**	**3**	**3**	6.15e-16	17.33915
f_8	ABC	**-3.86**	**-3.86**	**-3.86**	1.85e-13	27.04722
	HABCOB	**-3.86**	**-3.86**	**-3.86**	2.13e-15	**27.00847**
	HABCCH	**-3.86**	**-3.86**	**-3.86**	2.17e-15	26.86845
	HABCIC	**-3.86**	**-3.86**	**-3.86**	2.10e-15	27.34851
	HABCCOB	**-3.86**	**-3.86**	**-3.86**	**2.06e-15**	27.05978
	HABCO	**-3.86**	**-3.86**	**-3.86**	2.11e-15	27.53382
	HABC	**-3.86**	**-3.86**	**-3.86**	2.11e-15	27.39167

Table 5. Best, worst, mean, standard deviation and mean time value obtained by ABC and other six HABC through 25 independent runs on benchmark function f_9 - f_{16}.

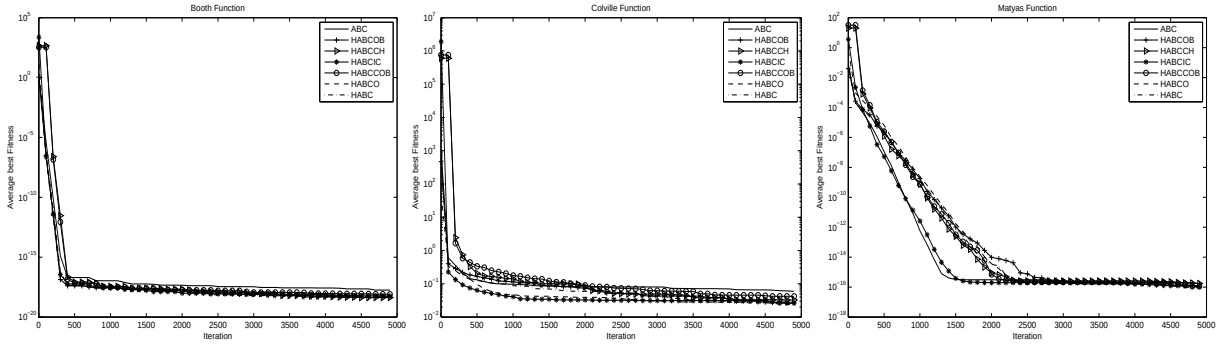
		Best	Mean	Worst	Std	Mean-Time
f_9	ABC	**4.65e-08**	**4.65e-08**	**4.65e-08**	**0**	**16.87600**
	HABCOB	**4.65e-08**	**4.65e-08**	**4.65e-08**	4.44e-17	17.20769
	HABCCH	**4.65e-08**	**4.65e-08**	**4.65e-08**	6.15e-17	17.04306
	HABCIC	**4.65e-08**	**4.65e-08**	**4.65e-08**	**0**	17.47723
	HABCCOB	**4.65e-08**	**4.65e-08**	**4.65e-08**	**0**	17.23549
	HABCO	**4.65e-08**	**4.65e-08**	**4.65e-08**	4.44e-17	17.51557
	HABC	**4.65e-08**	**4.65e-08**	**4.65e-08**	**0**	17.70736
f_{10}	ABC	8.27e-18	1.26e-16	4.59e-16	1.14e-16	**15.75365**
	HABCOB	1.32e-18	1.71e-16	2.81e-15	5.53e-16	16.18157
	HABCCH	1.00e-17	1.48e-16	4.87e-16	1.36e-16	15.91371
	HABCIC	5.29e-18	1.14e-16	3.51e-16	1.01e-16	16.42786
	HABCCOB	**6.36e-19**	1.12e-16	4.64e-16	1.21e-16	16.12124
	HABCO	1.42e-18	1.20e-16	5.32e-16	1.37e-16	16.62848
	HABC	1.85e-17	**8.90e-17**	**3.35e-16**	**9.44e-17**	16.51861
f_{11}	ABC	7.10e-03	5.20e-02	1.65e-01	4.50e-02	**22.32331**
	HABCOB	1.00e-03	1.39e-02	4.17e-02	1.15e-02	22.69221
	HABCCH	2.06e-03	1.32e-02	5.52e-02	1.09e-02	22.46052
	HABCIC	1.77e-03	**1.10e-02**	3.53e-02	8.59e-03	23.07073
	HABCCOB	2.16e-03	1.2e-02	3.92e-02	9.44e-03	22.81637
	HABCO	2.17e-03	1.13e-02	2.78e-02	**6.22e-03**	23.38827
	HABC	**4.54e-04**	1.18e-02	**3.40e-02**	9.32e-03	23.27893
f_{12}	ABC	8.98e-04	1.16e-03	1.38e-03	1.44e-04	**30.42611**
	HABCOB	1.30e-03	1.83e-03	2.82e-03	3.34e-04	31.00383
	HABCCH	6.60e-04	1.16e-03	2.49e-03	3.73e-04	30.83248
	HABCIC	**6.23e-04**	**7.28e-04**	**8.15e-04**	**4.80e-05**	31.27124
	HABCCOB	7.43e-04	1.14e-03	1.95e-03	2.51e-04	31.19106
	HABCO	7.69e-04	1.47e-03	2.39e-03	4. 00e-04	31.76304
	HABC	9.44e-04	1.83e-03	2.76e-03	3.81e-04	31.34683
f_{13}	ABC	**2.22e-04**	7.692e-03	1.83e-02	4.66e-03	**21.53706**
	HABCOB	4.57e-04	**3.906e-03**	1.66e-02	3.95e-03	21.93324
	HABCCH	3.30e-04	4.042e-03	1.84e-02	3.91e-03	21.94299
	HABCIC	4.12e-04	4.663e-03	1.34e-02	4.06e-03	22.28781
	HABCCOB	3.98e-04	5.589e-03	1.25e-02	3.75e-03	22.13560
	HABCO	4.59e-04	4.184e-03	**1.20e-02**	3.12e-03	22.40422
	HABC	2.78e-04	4.219e-03	9.70e-03	**3.01e-03**	22.26531
f_{14}	ABC	**-10.5364**	**-10.5364**	**-10.5364**	1.99e-15	**30.54940**
	HABCOB	**-10.5364**	**-10.5364**	**-10.5364**	2.76e-15	32.45616
	HABCCH	**-10.5364**	**-10.5364**	**-10.5364**	2.61e-15	31.69657
	HABCIC	**-10.5364**	**-10.5364**	**-10.5364**	2.76e-15	31.90898
	HABCCOB	**-10.5364**	**-10.5364**	**-10.5364**	3.03e-15	31.79735
	HABCO	**-10.5364**	**-10.5364**	**-10.5364**	2.90e-15	31.69738
	HABC	**-10.5364**	**-10.5364**	**-10.5364**	**1.36e-15**	31.69646
f_{15}	ABC	**-186.731**	**-186.731**	**-186.731**	1.00e-11	**21.36644**
	HABCOB	**-186.731**	**-186.731**	**-186.731**	2.84e-14	22.21063
	HABCCH	**-186.731**	**-186.731**	**-186.731**	3.01e-14	22.27073
	HABCIC	**-186.731**	**-186.731**	**-186.731**	2.17e-14	22.22425
	HABCCOB	**-186.731**	**-186.731**	**-186.731**	2.84e-14	22.28537
	HABCO	**-186.731**	**-186.731**	**-186.731**	**2.09e-14**	22.09370
	HABC	**-186.731**	**-186.731**	**-186.731**	3.62e-14	22.25839
f_{16}	ABC	**-50**	**-50**	**-50**	1.46e-13	**21.17621**
	HABCOB	**-50**	**-50**	**-50**	1.15e-13	22.05822
	HABCCH	**-50**	**-50**	**-50**	1.21e-13	22.15278
	HABCIC	**-50**	**-50**	**-50**	1.35e-13	22.16214
	HABCCOB	**-50**	**-50**	**-50**	1.57e-13	22.20180
	HABCO	**-50**	**-50**	**-50**	**1.08e-13**	22.03147
	HABC	**-50**	**-50**	**-50**	1.28e-13	22.19336

Table 6. Best, worst, mean, standard deviation and mean time value obtained by ABC and other six HABC through 25 independent runs on benchmark function f_{17} - f_{27} with dimension $D = 30$.

		Best	Mean	Worst	Std	Mean-Time
f_{17}	ABC	2.93e-14	3.81e-14	4.35e-14	3.85e-15	**21.70356**
	HABCOB	**2.58e-14**	3.16e-14	4.00e-14	2.69e-15	22.77789
	HABCCH	2.93e-14	3.27e-14	4.00e-14	2.40e-15	22.17633
	HABCIC	**2.58e-14**	3.16e-14	**3.29e-14**	2.02e-15	22.65846
	HABCCOB	2.93e-14	3.22e-14	4.00e-14	2.71e-15	22.41689
	HABCO	2.93e-14	**3.04e-14**	**3.29e-14**	**1.69e-15**	22.79281
	HABC	2.93e-14	3.06e-14	**3.29e-14**	1.74e-15	22.53482
f_{18}	ABC	1.63e-11	**1.43e-09**	**1.48e-08**	**3.29e-09**	**17.58065**
	HABCOB	1.89e-10	8.14e-06	1.85e-04	3.69e-05	18.01535
	HABCCH	1.16e-10	3.21e-08	2.65e-07	8.06e-08	18.21370
	HABCIC	1.13e-10	3.58e-06	2.30e-05	6.23e-06	18.48829
	HABCCOB	2.80e-11	5.27e-07	1.00e-05	2.00e-06	18.17515
	HABCO	4.21e-09	5.20e-08	4.13e-07	8.45e-08	18.71279
	HABC	**1.45e-11**	1.68e-07	1.38e-06	3.69e-07	18.40993
f_{19}	ABC	**0**	1.60e-16	4.44e-16	1.47e-16	**22.03209**
	HABCOB	**0**	3.55e-17	**1.11e-16**	5.29e-17	22.59067
	HABCCH	**0**	3.11e-17	**1.11e-16**	5.09e-17	22.88428
	HABCIC	**0**	3.11e-17	**1.11e-16**	5.09e-17	23.11554
	HABCCOB	**0**	3.11e-17	**1.11e-16**	5.09e-17	22.86108
	HABCO	**0**	4.88e-17	4.44e-16	9.66e-17	23.47623
	HABC	**0**	**2.66e-17**	**1.11e-16**	**4.84e-17**	23.05462
f_{20}	ABC	3.32e-16	5.02e-16	6.59e-16	7.45e-17	**32.90053**
	HABCOB	2.98e-16	4.16e-16	5.11e-16	6.35e-17	33.64184
	HABCCH	2.45e-16	4.10e-16	5.17e-16	8.13e-17	33.90082
	HABCIC	**1.99e-16**	**3.46e-16**	5.41e-16	7.81e-17	34.08160
	HABCCOB	2.86e-16	4.21e-16	5.07e-16	6.57e-17	33.89444
	HABCO	2.83e-16	4.13e-16	**4.93e-16**	**5.56e-17**	34.66688
	HABC	2.61e-16	4.03e-16	4.97e-16	6.34e-17	34.09351
f_{21}	ABC	-29.6275	-29.6176	-29.6059	6.20e-03	**25.27370**
	HABCOB	**-29.6309**	-29.6288	-29.6206	2.96e-03	25.95365
	HABCCH	**-29.6309**	-29.6271	-29.5882	8.69e-03	26.10902
	HABCIC	**-29.6309**	-29.6266	-29.5881	1.15e-02	26.29809
	HABCCOB	**-29.6309**	-29.6290	-29.6230	2.17e-03	26.17266
	HABCO	**-29.6309**	**-29.6302**	**-29.6273**	**8.40e-04**	26.57591
	HABC	**-29.6309**	-29.6298	-29.6254	1.62e-03	26.33754
f_{22}	ABC	**0**	6.82e-15	5.68e-14	1.89e-14	**18.69893**
	HABCOB	**0**	**0**	**0**	**0**	19.18100
	HABCCH	**0**	**0**	**0**	**0**	19.46644
	HABCIC	**0**	6.82e-15	5.68e-14	1.89e-14	19.58300
	HABCCOB	**0**	**0**	**0**	**0**	19.40121
	HABCO	**0**	**0**	**0**	**0**	19.84200
	HABC	**0**	**0**	**0**	**0**	19.53271
f_{23}	ABC	1.86e-04	1.34e-02	8.19e-02	2.38e-02	**18.19598**
	HABCOB	1.07e-04	1.41e-02	1.10e-01	2.48e-02	18.81008
	HABCCH	9.59e-05	1.07e-01	1.97e-00	3.93e-01	18.93122
	HABCIC	9.73e-04	9.59e-02	1.25e-00	2.51e-01	19.22760
	HABCCOB	6.02e-04	2.33e-01	3.67e-00	7.52e-01	18.93006
	HABCO	6.45e-03	1.38e-02	**2.13e-02**	**4.04e-03**	19.22689
	HABC	**8.18e-05**	**1.13e-02**	6.78e-02	1.78e-02	19.17352
f_{24}	ABC	**3.82e-04**	**3.82e-04**	**3.82e-04**	6.81e-13	**24.67135**
	HABCOB	**3.82e-04**	**3.82e-04**	**3.82e-04**	7.43e-13	25.42231
	HABCCH	**3.82e-04**	**3.82e-04**	**3.82e-04**	7.93e-13	25.00704
	HABCIC	**3.82e-04**	**3.82e-04**	**3.82e-04**	**6.03e-13**	25.22714
	HABCCOB	**3.82e-04**	**3.82e-04**	**3.82e-04**	1.34e-12	25.14297
	HABCO	**3.82e-04**	**3.82e-04**	**3.82e-04**	6.81e-13	25.49795
	HABC	**3.82e-04**	**3.82e-04**	**3.82e-04**	8.91e-13	25.46099
f_{25}	ABC	4.38e-16	5.05e-16	5.48e-16	3.35e-17	**22.23210**
	HABCOB	2.99e-16	4.22e-16	4.97e-16	6.54e-17	23.61736
	HABCCH	2.81e-16	4.19e-16	5.35e-16	6.70e-17	23.68333
	HABCIC	2.99e-16	4.10e-16	5.00e-16	6.85e-17	23.44918
	HABCCOB	2.41e-16	4.14e-16	5.03e-16	8.09e-17	23.86629
	HABCO	**1.21e-19**	**1.95e-16**	**4.84e-16**	**1.98e-16**	23.57562
	HABC	2.92e-16	4.08e-16	4.92e-16	6.91e-17	23.39717
f_{26}	ABC	3.06e-16	4.90e-16	5.51e-16	6.12e-17	**22.68787**
	HABCOB	2.87e-16	4.20e-16	5.21e-16	7.44e-17	23.93296
	HABCCH	3.03e-16	**3.81e-16**	**4.88e-16**	**5.84e-17**	24.04358
	HABCIC	2.61e-16	3.98e-16	5.00e-16	7.75e-17	23.84461
	HABCCOB	**2.50e-16**	3.82e-16	5.15e-16	7.86e-17	24.24357
	HABCO	2.92e-16	3.87e-16	5.05e-16	6.98e-17	24.04727
	HABC	3.05e-16	4.24e-16	5.15e-16	6.41e-17	23.89610
f_{27}	ABC	9.12e+01	1.68 e+02	2.35e+02	3.13e+01	**22.89231**
	HABCOB	1.16e+02	1.76e+02	2.41e+02	3.17e+01	24.36207
	HABCCH	1.83e+02	2.58e+02	3.52e+02	5.15e+01	24.08142
	HABCIC	1.68e+02	2.68e+02	3.30e+02	4.91e+01	24.03064
	HABCCOB	1.50e+02	2.52e+02	3.89e+02	6.66e+01	24.19654
	HABCO	**1.59e-00**	**3.38e-00**	**8.77e-00**	**1.78e-00**	25.14578
	HABC	1.01e+02	1.93e+02	2.52e+02	3.94e+01	24.23884

Table 7. Best, worst, mean, standard deviation and mean time value obtained by ABC and other six HABC through 25 independent runs on benchmark function f_{17} - f_{27} with dimension $D=100$.

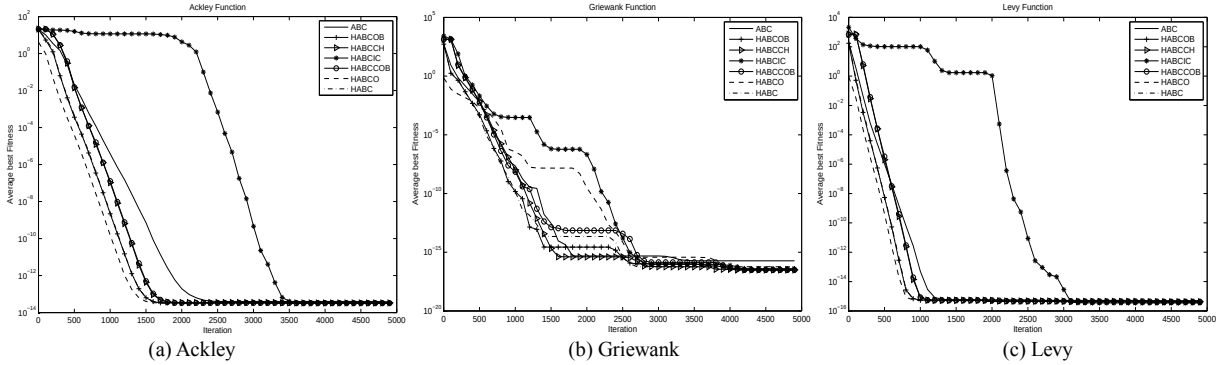
		Best	Mean	Worst	Std	Mean-Time
f_{17}	ABC	8.70e-10	2.16e-09	4.23e-09	9.00e-10	**27.42205**
	HABCOB	2.71e-13	3.14e-13	3.53e-13	2.51e-14	28.17576
	HABCCH	4.70e-13	7.05e-13	1.34e-12	1.75e-13	28.39553
	HABCIC	9.01e-00	1.17e+01	1.35e+01	1.20e-00	29.17666
	HABCCOB	5.84e-13	6.68e-13	7.61e-13	5.31e-14	28.36998
	HABCO	**1.47e-13**	**1.65e-13**	**1.75e-13**	**8.47e-15**	40.89752
	HABC	2.60e-13	3.19e-13	3.77e-13	3.30e-14	27.96116
f_{18}	ABC	3.91e-04	**1.69e-03**	**4.65e-03**	1.10e-03	**26.87399**
	HABCOB	2.58e-04	6.20e-01	7.14e-00	1.57e-00	27.88947
	HABCCH	4.59e-04	7.44e-01	1.01e+01	2.25e-00	27.87438
	HABCIC	1.29e-02	4.50e-00	1.73e+01	5.08e-00	27.26161
	HABCCOB	5.48e-04	5.45e-02	1.15e-00	2.28e-01	28.08510
	HABCO	6.67e-01	6.67e-01	6.68e-01	**2.33e-04**	39.36317
	HABC	**1.04e-04**	5.56e-01	4.90e-00	1.31e-00	27.74415
f_{19}	ABC	9.99e-16	5.21e-15	4.81e-14	9.27e-15	**29.54152**
	HABCOB	9.99e-16	1.20e-13	2.66e-12	5.30e-13	31.11756
	HABCCH	5.55e-16	9.18e-11	2.30e-09	4.59e-10	31.43440
	HABCIC	**4.44e-16**	6.56e-13	1.48e-11	2.96e-12	31.06522
	HABCCOB	8.88e-16	2.06e-11	5.10e-10	1.02e-10	31.43196
	HABCO	4.11e-15	1.20e-07	2.89e-06	5.77e-07	48.57922
	HABC	9.99e-16	**4.20e-15**	**1.89e-14**	**4.98e-15**	30.98284
f_{20}	ABC	2.53e-15	3.14e-15	3.66e-15	2.95e-16	**62.18824**
	HABCOB	2.29e-15	2.65e-15	3.17e-15	2.30e-16	63.43297
	HABCCH	2.30e-15	2.67e-15	3.13e-15	2.02e-16	63.48975
	HABCIC	4.26e-06	1.95e+01	1.02e+02	3.18e+01	62.75135
	HABCCOB	**2.10e-15**	**2.62e-15**	**2.98e-15**	1.86e-16	63.74450
	HABCO	2.53e-15	2.78e-15	3.12e-15	1.89e-16	84.86471
	HABC	2.30e-15	2.80e-15	3.20e-15	**1.72e-16**	63.15849
f_{21}	ABC	-98.3566	-97.9593	-97.4569	2.52e-01	40.77573
	HABCOB	-98.9313	-98.5219	-98.249	1.67e-01	41.93359
	HABCCH	-98.5658	-98.1583	-97.7409	1.84e-01	41.98199
	HABCIC	-97.8113	-92.8133	-88.6395	2.29e-00	**40.44041**
	HABCCOB	-98.5910	-98.1587	-97.4723	2.31e-01	42.11596
	HABCO	-98.9559	-98.7545	-98.5844	9.32e-02	58.16782
	HABC	-98.7921	-98.4558	-98.1394	1.55e-01	41.84345
f_{22}	ABC	3.41e-13	5.34e-09	9.09e-08	1.85e-08	**27.92113**
	HABCOB	**2.27e-13**	5.64e-13	1.48e-12	2.96e-13	29.00650
	HABCCH	3.41e-13	4.17e-04	6. 81e-03	1.52e-03	28.94996
	HABCIC	4.98e+01	1.55e+02	2.51e+02	5.73e+01	28.52347
	HABCCOB	**2.27e-13**	1.50e-03	3.71e-02	7.42e-03	28.95015
	HABCO	**2.27e-13**	5.14e-13	**7.96e-13**	**1.47e-13**	42.21861
	HABC	**2.27e-13**	**4.91e-13**	9.09e-13	1.82e-13	28.88114
f_{23}	ABC	**7.78e-03**	**6.16e-02**	**1.78e-01**	**4.09e-02**	27.95234
	HABCOB	3.25e-02	6.98e-01	3.18e-00	8.34e-01	29.33051
	HABCCH	1.07e-02	1.09e+01	8.30e+01	2.47e+01	28.84404
	HABCIC	7.75e+01	1.26e+02	1.99e+02	3.26e+01	**27.48526**
	HABCCOB	5.56e-03	3.36e-00	3.87e+01	7.76e-00	29.10317
	HABCO	3.70e-02	1.23e-01	4.05e-01	1.19e-01	41.59810
	HABC	1.76e-02	5.71e-01	4.72e-00	9.86e-01	29.22457
f_{24}	ABC	2.02e-03	3.87e+02	8.29e+02	2.09e+02	**25.57455**
	HABCOB	**1.27e-03**	**1.27e-03**	1.28e-03	2.20e-06	27.92126
	HABCCH	3.20e+02	8.77e+02	1.42e+03	2.93e+02	26.63203
	HABCIC	**1.27e-03**	9.53e-00	1.18e+02	3.28e+01	27.17422
	HABCCOB	2.37e+02	9.42e+02	1.66e+03	3.49e+02	26.62176
	HABCO	**1.27e-03**	**1.27e-03**	**1.27e-03**	**3.38e-11**	43.24939
	HABC	**1.27e-03**	3.61e-03	5.97e-03	1.17e-02	27.79104
f_{25}	ABC	2.76e-15	3.24e-15	3.63e-15	2.23e-16	**24.52196**
	HABCOB	2.52e-15	2.80e-15	3.12e-15	**1.68e-16**	25.71166
	HABCCH	**1.88e-15**	2.69e-15	3.20e-15	2.85e-16	25.88816
	HABCIC	1.98e-15	**2.65e-15**	**2.98e-15**	2.59e-16	25.58481
	HABCCOB	2.32e-15	2.72e-15	2.99e-15	1.79e-16	25.91885
	HABCO	2.19e-15	2.76e-15	3.17e-15	1.98e-16	37.45111
	HABC	2.26e-15	2.66e-15	3.20e-15	2.51e-16	25.60685
f_{26}	ABC	2.50e-15	3.19e-15	3.77e-15	3.78e-16	**24.92382**
	HABCOB	**2.02e-15**	2.68e-15	3.20e-15	2.34e-16	26.53891
	HABCCH	2.06e-15	2.72e-15	3.17e-15	2.55e-16	26.69612
	HABCIC	2.25e-15	2.70e-15	2.99e-15	2.11e-16	26.46492
	HABCCOB	2.10e-15	**2.63e-15**	2.98e-15	2.17e-16	26.83566
	HABCO	2.28e-15	2.64e-15	3.18e-15	2.04e-16	38.26213
	HABC	2.07e-15	2.69e-15	**2.97e-15**	**2.02e-16**	26.41392
f_{27}	ABC	1158.267	1283.235	1372.793	**4.38e+01**	**25.23831**
	HABCOB	1212.111	1342.483	1470.248	5.96e+01	27.00475
	HABCCH	1696.155	1913.789	2113.579	1.17e+02	26.95500
	HABCIC	1189.182	1361.141	1476.671	7.33e+01	26.82238
	HABCCOB	1542.838	1873.756	2065.895	1.39e+02	26.88027
	HABCO	1130.100	1367.226	1476.271	7.73e+01	38.83916
	HABC	1220.647	1323.562	1466.346	6.51e+01	26.65520



(a) Booth (b) Colville (c) Matyas

(d) Perm (e) Powell (f) PowerSum

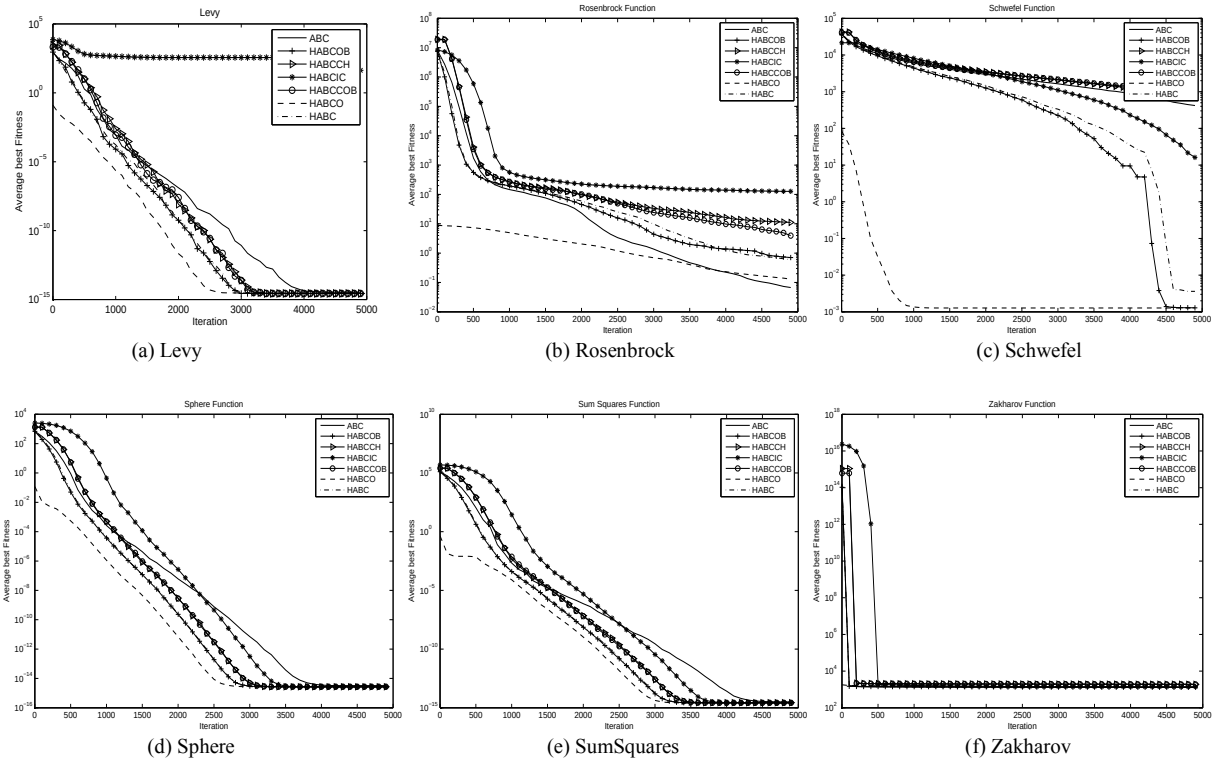
Figure 1. Mean of best function values for test functions.



(a) Ackley (b) Griewank (c) Levy

(d) Sphere (e) SumSquares (f) Zakharov

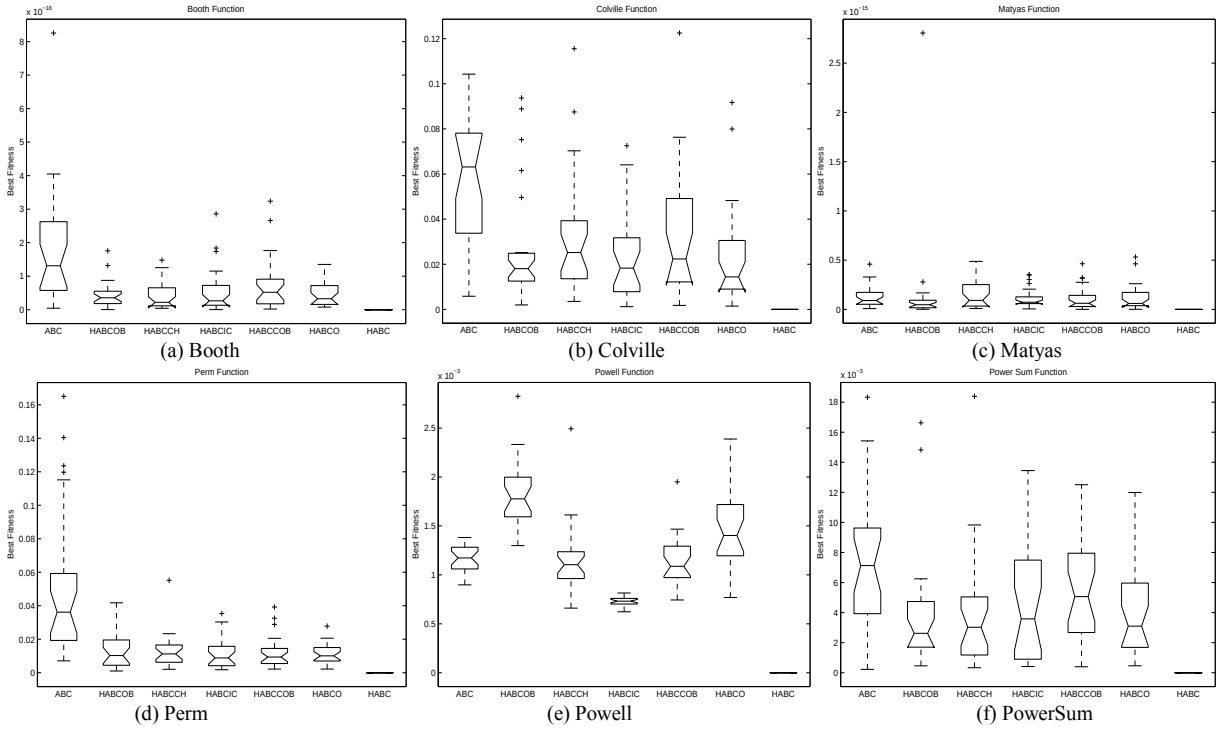
Figure 2. Mean of best function values for test functions with dimension $D = 30$.



(a) Levy (b) Rosenbrock (c) Schwefel

(d) Sphere (e) SumSquares (f) Zakharov

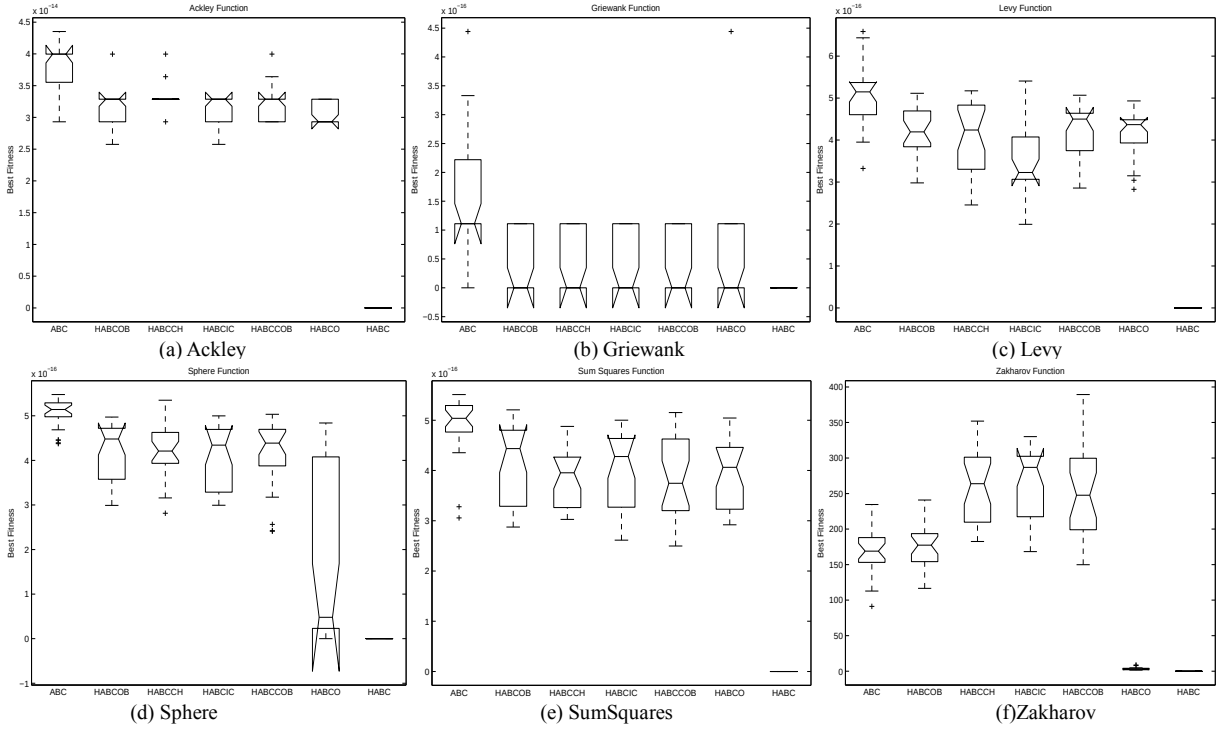
Figure 3. Mean of best function values for test functions with dimension $D = 100$.



(a) Booth (b) Colville (c) Matyas

(d) Perm (e) Powell (f) PowerSum

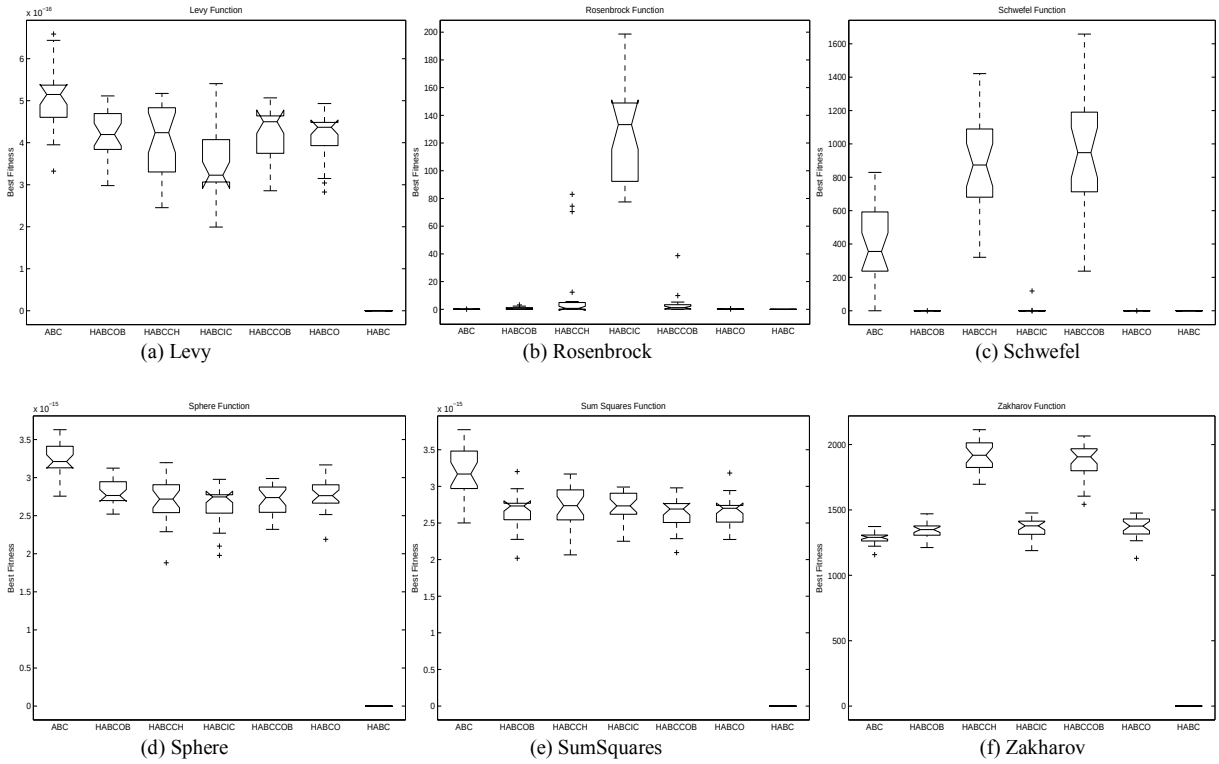
Figure 4. Statistical values of the function values for test functions.



(a) Ackley (b) Griewank (c) Levy

(d) Sphere (e) SumSquares (f)Zakharov

Figure 5. Statistical values of the function values for test functions with dimension $D = 30$.



(a) Levy (b) Rosenbrock (c) Schwefel

(d) Sphere (e) SumSquares (f) Zakharov

Figure 6. Statistical values of the function values for test functions with dimension $D = 100$.

[3] M. Dorigo, T. Stutzle. Ant colony optimization. Cambridge: MAMIT Press, 2004.
[4] R. Storn, K. Price. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 1997, 11(4):341-359.
[5] J. Kennedy, R. Eberhart. Particle swarm optimization. In IEEE International Conference on Neural Networks, 1995, pp. 1942-1948.
[6] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Kayseri, Turkey: ErciyesUniversity, 2005.
[7] D. Karaboga, B.Basturk. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation, 2009, 214(1):108-132. [8] A. Singh. An artificial bee colony algorithm for the leaf constrained minimum spanning tree problem. Applied Soft Computing Journal, 2009, 9(2):625-631.
[9] Q. K. Pan, M. F. Tasgetiren, P. Suganthan, T. Chua. A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Information Sciences, 2011, 181(12):2455-2468.
[10] F. Kang, J. Li, Q. Xu. Structural inverse analysis by hybrid simplex artificial bee colony algorithms. Computers and Structures, 2009, 87(13):861-870.
[11] R. S. Rao, S. V. L. Narasimham, M. Ramalingaraju. Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm. International Journal of Electrical Power and Energy Systems Engineering, 2008, 1(2):116-122.
[13] C. S. Zhang, D. T. Ouyang, J. X. Ning. An artificial bee colony approach for clustering. Expert Systems with Application, 2010, 37(7):4761-4767.
[14] Z. Hu, M. Zhao. Simulation on traveling salesman problem (TSP) based on artificial bees colony algorithm. Transaction of Beijing Institute of Technology, 2009, 29(11):978-982.
[15] Y. W. Leung, Y. P. Wang. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions of Evolutionary Computation, 2001, 5(1):41-53.
[16] B. Alatas. Chaotic bee colony algorithms for global numerical optimization. Expert Systems with Applications, 2010, 37(8):5682-5687.
[17] S. Rahnamayan, et al. Opposition-based differential evolution. IEEE Transaction on Evolutionary Computation, 2008, 12(1):64–79.
[18] W. Gao, S. Liu. A modified artificial bee colony algorithm. Computers and Operations Research, 2012, 39(3):687-697.
[19] Y. W. Leung, Y. P. Wang. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Transactions of Evolutionary Computation, 2001, 5(1):41-53.
[20]X. Kong, et al. . Hybrid Artificial Bee Colony Algorithm for Global Numerical Optimization. Journal of Computational Information System, 2012, 8(6):2367-2374.

Xiangyu Kong received his B.A. in Applied Mathematics from Hunan University, Changsha, China, in 2004, the M.S. degree in Applied Mathematics from Xidian University, Xi'an, China, in 2009. Since the year of 2011, he started his learning in Applied Mathematics from Xidian University for his Ph. D. degree. He joined Department of Applied Mathematics, Zhoukou Normal University, China, in 2004 as a research/teaching assistant and then became Instructor in the same department. His main interests include Intelligent Optimization, Theory and Methods of Optimization.

Sanyang Liu is a professor and a tutor of Ph.D. in Xidian University. His current research interests include optimization theory and algorithm, machine learning and pattern recognition.

Zhen Wang lecturer received her Ph.D. degree in Applied Mathematics from Xidian University in 2012. She is currently with Institute of Information and System Computation Science, Beifang University of Nationalities, China. Her main research areas include Mathematical Finance, Portfolio Optimization, and Intelligent Optimization.