

Archimedean copula estimation of distribution algorithm based on artificial bee colony algorithm

Haidong Xu, Mingyan Jiang*, and Kun Xu

School of Information Science and Engineering, Shandong University, Jinan 250100, China

Abstract: The artificial bee colony (ABC) algorithm is a competitive stochastic population-based optimization algorithm. However, the ABC algorithm does not use the social information and lacks the knowledge of the problem structure, which leads to insufficiency in both convergent speed and searching precision. Archimedean copula estimation of distribution algorithm (ACEDA) is a relatively simple, time-economic and multivariate correlated EDA. This paper proposes a novel hybrid algorithm based on the ABC algorithm and ACEDA called Archimedean copula estimation of distribution based on the artificial bee colony (ACABC) algorithm. The hybrid algorithm utilizes ACEDA to estimate the distribution model and then uses the information to help artificial bees to search more efficiently in the search space. Six benchmark functions are introduced to assess the performance of the ACABC algorithm on numerical function optimization. Experimental results show that the ACABC algorithm converges much faster with greater precision compared with the ABC algorithm, ACEDA and the global best (gbest)-guided ABC (GABC) algorithm in most of the experiments.

Keywords: artificial bee colony (ABC) algorithm, Archimedean copula estimation of distribution algorithm (ACEDA), ACEDA based on artificial bee colony (ACABC) algorithm, numerical function optimization.

DOI: 10.1109/JSEE.2015.00045

1. Introduction

Population-based optimization algorithms are biological-inspired optimization algorithms which are capable of finding the near-optimal solutions to complicated numerical and real-valued problems. Some classic algorithms such as the genetic algorithm (GA) [1], the estimation of distribution algorithm (EDA) [2] and the artificial bee colony (ABC) algorithm [3] are all population-based optimization

algorithms.

The ABC algorithm was proposed by Karaboga in 2005 [3], which is inspired by the foraging behavior of bees. The ABC algorithm is simple and uses only common control parameters such as the population size and the max cycle number. Experimental results on function optimization have shown that the ABC algorithm outperforms, or at least performs as well as other well known population-based algorithms like GA, PSO [4,5], etc.

However, there are still insufficiencies in the ABC algorithm. The population reproduction mechanism of the ABC algorithm reveals that this algorithm does not take advantage of the social information such as the global best value of the current swarm, and lacks the knowledge of the problem structure when searching for the optimal value, which leads to insufficiency in both convergent speed and searching precision. Moreover, the solution search equation reveals that the ABC algorithm is good at exploration but poor at exploitation. To conquer this problem, various improved strategies have been proposed. Zhu and Kwong [6] proposed global best (gbest)-guided ABC (GABC) by incorporating the information of the gbest solution into the solution search equation to improve the exploitation. Lei et al. [7] proposed an improved ABC algorithm, which introduced an inertial weight to the original ABC iteration equation to balance local and global searching processes. Akay and Karaboga [8] modified the original ABC algorithm by employing two new solution search strategies, including frequency and magnitude of the perturbation. Gao and Liu [9] improved the ABC algorithm by combining the mutation scheme of the differential evolution (DE). Experimental results [6–9] show that these variant ABC algorithms improved the convergent speed and the searching precision to a certain extent.

Estimation of distribution algorithm (EDA) is an evolutionary algorithm which derives from GA [10], and it combines intelligence computation and the knowledge of statistics. EDA uses the probability modeling technique to

Manuscript received on March 19, 2014.

*Corresponding author.

This work was supported by the National Natural Science Foundation of China (61201370), the Special Funding Project for Independent Innovation Achievement Transform of Shandong Province (2012CX30202) and the Natural Science Foundation of Shandong Province (ZR2014FM039).

guide the generation of new population. Population-based incremental learning (PBIL) proposed by Baluja [11], the Bayesian optimization algorithm (BOA) proposed by Pelikan [12] are all popular EDAs.

The copula theory was introduced into EDA by Fabrizio and Carlo [13]. In copula EDA (CEDA), joint distribution of all variables is used to describe the correlation among variables, which can estimate the problem structure in a simple way. Archimedean copula EDA (ACEDA) is an important branch in CEDA, and a number of researches have been done in this field [14,15].

In this paper, we propose a new hybrid algorithm based on the ABC algorithm and the ACEDA, which is called Archimedean copula ABC (ACABC) algorithm. This hybrid algorithm utilizes the ACEDA to estimate the distribution model (that is, to learn the problem structure) and then uses the information to guide artificial bees to search more efficiently in the search space. This hybrid algorithm combines the advantages of both ABC and ACEDA. On one hand, ACABC keeps learning the problem structure during the searching process by incorporating the ACEDA mechanism, and uses the information to generate new population. This mechanism can guide bees to search directly, which accelerates the convergent speed and improves the exploration ability. On the other hand, an improved gbest-guided mechanism is introduced into ACABC to improve the exploitation ability. The experimental results tested on six numerical benchmark functions show that the ACABC algorithm converges much faster with greater precision compared with the ABC algorithm, ACEDA and the GABC algorithm in most of the experiments.

The rest of the paper is organized as follows. Section 2 summarizes the ABC algorithm. Section 3 introduces the basic copula theory and ACEDA. The hybrid ACABC algorithm is described in Section 4 and experimental settings and results are given in Section 5. Finally, the conclusion is drawn in Section 6.

2. ABC algorithm

As mentioned above, ABC is a swarm intelligence algorithm by simulating the foraging behaviors of the bee swarm. In a natural bee swarm, there are three kinds of bees to search food, the employed bees, the onlookers, and the scouts. The employed bees search food around the food source in their memories, then they share the food information with the onlookers through waggle dancing. Each onlooker bee chooses a food source found by the employed bees, and then further searches around the selected food source. The food source with more nectar has larger chance to be selected by the onlooker bees than the one with less nectar. The scouts are a few employed bees which abandon their original food sources and randomly search for new ones.

For an optimization problem in a D -dimensional space, the position of a food source represents a potential solution, and the nectar amount of a food source represents the fitness value of the solution. The number of employed bees or the onlooker bees is equal to the number of food sources, which means one food source is exploited by one employed bee.

$\mathbf{X}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ denotes the i th food source in the population, where D is the dimension of the problem. The exploitation mechanism used by both the employed bees and the onlookers is given as follows:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (1)$$

where $\mathbf{V}_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ is the new candidate solution generating from the neighborhood of current solution \mathbf{X}_i ($i = 1, 2, \dots, N$), N is the population size, \mathbf{X}_k is a randomly selected solution in the population ($k = 1, 2, \dots, N$ and $k \neq i$), $j = 1, 2, \dots, D$ is a random index, and ϕ_{ij} is a uniform random number in the range $[-1, 1]$. Then the greedy selection is operated between \mathbf{V}_i and \mathbf{X}_i to retain the better solution.

When all the employed bees finish their neighborhood search according to (1), they share all the food information with all the onlookers. Each onlooker selects a food source to do further search according to the probability calculated by (2).

$$p_i = \frac{fit_i}{\sum_{j=1}^N fit_j} \quad (2)$$

where fit_i is the fitness value of the i th solution in the population, and p_i is called the following probability. As shown in (2), p_i is proportional to the fitness value, and the solution with a larger fitness value has a higher chance to be selected.

If one food source is not updated over a predefined number of cycles, which means there is no better food source in its neighborhood, this food source is abandoned by the employed bee. The predefined number of cycles is a control parameter called “limit”. When the employed bee abandons its food source, it becomes a scout and searches a new food source randomly in the whole searching space according to (3).

$$x_{ij} = x_j^{\min} + rand(0, 1)(x_j^{\max} - x_j^{\min}) \quad (3)$$

where $[x_j^{\min}, x_j^{\max}]$ is the boundary of the j th variable, and $rand(0, 1)$ is a uniformly distributed random number between 0 and 1.

3. ACEDA

EDA is an evolutionary algorithm derived from GA [16], which combines intelligence computation and the knowl-

edge of statistics. EDA retains the selection operator in GA, but replaces the crossover and the mutation operator with the statistical model and the sampling theory. One of the most essential parts in EDA is to build a proper statistical model.

The copula theory is a new branch in statistics, which constructs a multivariate joint distribution function with a given marginal distribution function and correlations among all variables. Basic definitions and theorems are given in [16]. One essential theorem in the copula theory

is the Sklar theorem [16]. This theorem expounds the construction method of multivariate joint distribution by using the copula function and marginal distribution functions. In the copula theory, the multivariate joint distribution function is constructed based on the essential Sklar theorem.

Copula functions mainly includes elliptic functions [17] and Archimedean copula functions [18]. Definitions of the Archimedean copula function are given in [19]. Two types of Archimedean copula functions Clayton and Gumbel are mainly used in this paper, as is shown in Table 1.

Table 1 Clayton and Gumbel function

Name	Generator	Copula function	Parameter
Clayton	$\varphi(t) = t^{-\theta} - 1$	$C = (1 + \sum_{i=1}^n (u_i^{-\theta} - 1))^{-1/\theta}$	$\theta \geq -1, \theta \neq 0$
Gumbel	$\varphi(t) = (-\ln t)^\theta$	$C = \exp(-(\sum_{i=1}^n (-\ln u_i)^\theta)^{1/\theta})$	$\theta \geq 1$

ACEDA is based on EDA and the copula theory, which constructs the statistical model with the Archimedean copula function and the marginal distribution function based on the Sklar theorem. In this algorithm, the estimation of the statistical model includes copula function estimation and marginal function estimation, then new population is generated by sampling from the copula function. The framework of ACEDA is shown in Fig. 1.

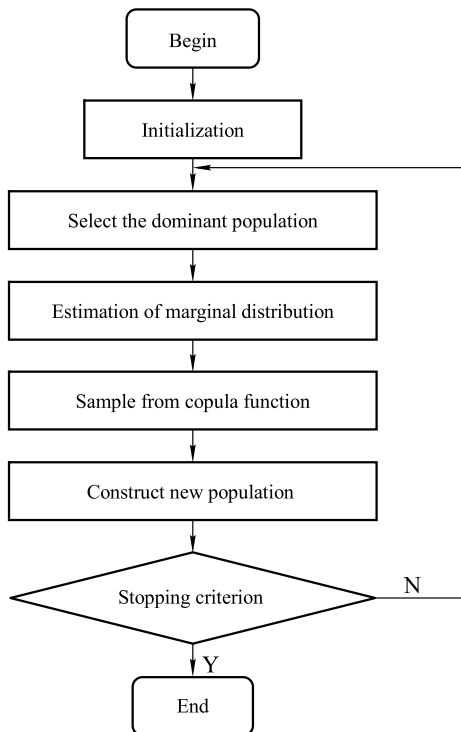


Fig. 1 Framework of ACEDA

In the following, the estimation of the marginal distribution function and the sampling method of the n -

dimensional Archimedean copula function are first introduced, then the specific process of the Archimedean copula EDA is listed.

In the n -dimensional ACEDA, there are mainly two marginal distribution estimation methods. One is based on the empirical function, and the other is based on the Gaussian probability model. In this paper, the Gaussian probability model is used to estimate the marginal distribution. After selecting the dominant population which consists of S individuals, mean value μ_j and standard deviation σ_j of the j th dimension variable are calculated according to (4) and (5) respectively. The j th Gaussian marginal distribution is denoted as $N(\mu_j, \sigma_j^2)$ ($j = 1, 2, \dots, n$).

$$\mu_j = \frac{1}{S} \sum_{i=1}^S x_{ij} \quad (4)$$

$$\sigma_j = \sqrt{\frac{1}{S} \sum_{i=1}^S (x_{ij} - \mu_j)^2} \quad (5)$$

The sampling method of the n -dimensional Archimedean copula function is as follows. C denotes the Archimedean copula function, and φ represents its generator, while (U_1, U_2, \dots, U_n) is the random vector which obeys the joint distribution C . According to the algorithm proposed by Marshall and Olkin in [20], if there is a distribution function F which yields that $F(0) = 0$, and the Laplace transform of F is equal to the inverse function of generator φ , that is to say $\varphi^{-1} = L^{-1}[F]$, then the samples (u_1, u_2, \dots, u_n) of (U_1, U_2, \dots, U_n) can be generated as follows.

Algorithm 1 The sampling method of the n -dimensional Archimedean copula function

Step 1 Generate variable v which obeys the distribution F , $v \sim F = L^{-1}[\varphi^{-1}]$, where $L^{-1}[\varphi^{-1}]$ denotes the inverse Laplace transform of φ^{-1} .

Step 2 Generate independent variables $x_j \sim U[0, 1]$, $j = 1, 2, \dots, n$;

Step 3 $u_j = \varphi^{-1}((-\ln x_j)/v)$, $j = 1, 2, \dots, n$, then the sample (u_1, u_2, \dots, u_n) which obeys the joint distribution function C is generated.

The process of Clayton and Gumbel ACEDA is shown as follows.

Algorithm 2 The Clayton and Gumbel ACEDA

Step 1 Population initialization. Initialize the population size NP , iterations $Cycle$, selection ratio s , and mutation ratio c . Randomly generate population size solutions as the initial population, and then calculate the fitness value of each population.

Step 2 Construct dominant population. Sort the population in descending order according to the fitness values; select the top S individuals according to (6) to construct the dominant population.

$$S = sNP \quad (6)$$

Step 3 Estimate the marginal Gaussian distribution of each dimension $N(\mu_j, \sigma_j^2)$ ($j = 1, 2, \dots, n$), according to (4) and (5).

Step 4 Perform sampling operation on the given copula function to generate L new individuals which obey the joint distribution.

Step 4.1 The inverse Laplace transform of φ^{-1} for the Clayton and Gumbel function obeys Gamma distribution and Alpha-stable distribution respectively [21], as is shown in (7) and (8). Generate variable v which subjects to Gamma distribution or Alpha-stable distribution according to (7) or (8). Generate independent variables v_j ($j = 1, 2, \dots, n$), which is uniformly distributed in the range $[0, 1]$, and then u_j ($j = 1, 2, \dots, n$), for the Clayton or Gumbel function which is obtained according to (9) or (10).

$$L^{-1}[\varphi^{-1}] = F(v) = \frac{v^{\frac{1}{\theta}-1} e^{-\frac{v}{\theta}}}{\theta^{\frac{1}{\theta}} \Gamma\left(\frac{1}{\theta}\right)} \sim \text{Gamma}\left(\frac{1}{\theta}, \theta\right) \quad (7)$$

$$F(v) \sim \begin{cases} \text{stable}\left(\frac{1}{\theta}, 1, \left(\cos\left(\frac{\pi\alpha}{2}\right)\right)^{\frac{1}{\alpha}} / \exp\left(\frac{\pi}{2}\right), 0\right), & \alpha \neq 1 \\ \text{stable}\left(\frac{1}{\theta}, 1, 0, 1\right), & \alpha = 1 \end{cases} \quad (8)$$

$$u_j = \varphi^{-1}((-\ln v_j)/v) = \left(1 - \frac{\theta \ln v_j}{v}\right)^{-\frac{1}{\theta}} \quad (9)$$

$$u_j = \varphi^{-1}((-\ln v_j)/v) = \exp\{-[(-\ln v_j)/v]^{\frac{1}{\theta}}\} \quad (10)$$

Step 4.2 According to (11), generate the k th new individual $(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})$ based on the combination of the one-dimension Gaussian distribution and the Clayton or Gumbel copula function value.

$$x_j^{(k)} = F_j^{-1}(u_j; \mu_j; \sigma_j) \quad (11)$$

Step 5 Construct new population. The new population consists of the top S individuals from the previous generation, L new individuals which subject to the joint distribution and $NP - S - L$ mutated individuals which are randomly generated in the searching space.

Step 6 Judge whether the algorithm satisfies the terminal condition or not. If the terminal condition is reached, the program will stop. Otherwise, return to Step 2.

4. ACABC algorithm

In this paper, we propose an improved ABC algorithm by combining ABC with ACEDA to improve the convergent speed and searching precision. The novel hybrid algorithm is ACABC algorithm.

In the ABC algorithm, the onlookers obtain food information from all employed bees, choose a better food source and then do further search around it. The onlooker mechanism is one of the most essential mechanisms in the ABC algorithm. In the ACABC algorithm, we combine the ACEDA with onlookers and propose a modified onlooker mechanism. After obtaining all the food information, the modified onlookers select excellent food sources with a certain ratio, analyze the distribution rule of excellent food sources by estimating the distribution of them. Then these onlookers take the following strategies to update the current food sources. Firstly, preserve the excellent food sources selected previously, then predict the location of new food sources according to the estimated distribution information, and finally replace those poor food sources with these predicted new ones. That is to say, the onlookers adjust the searching direction of the whole colony based on the distribution of promising food sources. This modified onlooker mechanism has learning and analyzing ability and enhanced global performance, which is a more intelligent onlooker mechanism. The implementation flow of this modified onlooker mechanism is shown as follows.

Algorithm 3 Modified onlooker mechanism

Step 1 Sort the solutions obtained by employed bees in descending order based on the fitness values, and calculate the following probability of each solution according to (2).

Step 2 Set the selection ratio $s = 0.3$, and select the top S solutions according to (6) to construct the dominant population.

Step 3 Estimate the marginal Gaussian distribution of each variable according to (4) and (5).

Step 4 After choosing one solution according to the following probability, the onlooker decides the sort order of this solution. If the sort order is less than S , then the neighborhood search mechanism of ABC is operated to generate new candidate solution; otherwise, the new candidate solution is generated based on ACEDA.

Step 5 Calculate the fitness value of new solutions, choose a better solution between the new candidate solution and the old one based on the greedy criterion to construct the new population.

On the basis of the modified onlooker mechanism, two more improved strategies are proposed to enhance the adaptability and the exploitation.

4.1 Dynamical adjustment of the searching strategy

The modified onlooker mechanism shown above includes two searching strategies. According to the selection ratio $s = 0.3$, the top 30% individuals are updated by the neighborhood search strategy while the rest 70% are updated based on ACEDA. However, as the iteration goes on, an increasing number of individuals are approaching the global best solution and the ratio of excellent individuals is getting larger. As a result, a growing number of individuals can not be updated by ACEDA, which leads to the searching inefficiency. To improve this situation, an adaptive selection ratio is introduced, as is shown in (12).

$$s = 0.3 + 0.7 \times \frac{iter}{Cycle} \quad (12)$$

$$S = \lfloor sNP \rfloor \quad (13)$$

where $iter$ denotes the current iteration number, and $Cycle$ is the maximal iteration number. The initial value is set as 0.3, and the selection ratio grows linearly as the iteration increases. The number of selected individuals in each iteration is calculated according to (13), which indicates that more individuals are updated by the neighborhood searching mechanism while fewer of them are updated by copula EDA. In a word, the searching strategies are adjusted dynamically by introducing the adaptive parameter to improve the search efficiency.

4.2 Improved gbest-guided neighborhood search mechanism

Inspired by the gbest-guided mechanism in [6], the neighborhood search equation is modified to improve the exploitation of the algorithm, as is shown in (14).

$$v_{ij} = wx_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + (1 - w)(y_j - x_{ij}) \quad (14)$$

$$w = 0.5 \times (0.9 - 0.6 \times \frac{iter}{Cycle}) \quad (15)$$

where the third term is a new added term, and y_j is the j th element of the current global best solution. As it is in (15), a new adaptive parameter w is introduced to adjust the neighborhood search strategy dynamically. w decreases linearly as the iteration goes on, which means that the generation of the new candidate solution is increasingly dependent on the global best value, while the influence of the current solution x_{ij} is weakened progressively. The new candidate solution is driven towards the global best solution in this way to improve the exploitation.

The process of the ACABC algorithm is shown as follows.

Algorithm 4 Archimedean copula estimation of distribution by the ABC algorithm

Step 1 Parameter initialization. Initialize the population size NP , iterations $Cycle$, $trial$, $limit$, selection ratio s , and the parameter θ used in ACEDA.

Step 2 Population initialization. Randomly generate NP solution $\mathbf{X} = \{\mathbf{X}_i | i = 1, 2, \dots, NP\}$ as the initial population, calculate the fitness value of each population, set $trial_i = 0$ for each solution \mathbf{X}_i and then initialize the gbest solution.

Step 3 Employed bee stage. Each employed bee conducts the gbest-guided neighborhood search according to (14) to generate the new candidate solution, calculates the fitness value, and updates the current solution based on the greedy criterion. If \mathbf{X}_i is updated as the new candidate solution, set $trial_i = 0$; otherwise, $trial_i = trial_i + 1$.

Step 4 Calculate the following probability according to (2), sort all solutions in descending order according to fitness values, compute the number of excellent individuals S according to (12) and (13), and select the top S individuals to construct the dominant population.

Step 5 The onlooker stage. Carry out Step 3 – Step 5 in Algorithm 3.

Step 6 Update the global best solution. Select the solution with the largest fitness value as the current gbest solution.

Step 7 The scouts stage. Give up the solution \mathbf{X}_i with $trial_i$ exceeding the predefined value $limit$, randomly generate a new solution in the search space, calculate the fitness value, and set $trial_i = 0$.

Step 8 Judge whether the algorithm satisfies the terminal condition or not. If the terminal condition is reached, output the global best solution as the final optimization result. Otherwise, return to Step 3.

5. Experiments and results

In this paper, two types of ACABC algorithms, namely the Clayton ACABC algorithm and the Gumbel ACABC algorithm are proposed based on Archimedean Clayton and Gumbel functions. In this part, the performance of the Clayton ACABC algorithm and the Gumbel ACABC al-

gorithm are compared with ABC, GABC, Clayton CEDA and Gumbel CEDA by optimizing benchmark functions.

5.1 Benchmark functions

The details of six benchmark functions used in this paper are given in Table 2.

Table 2 Benchmark functions

Test function	$f(x)$	Search range	x^*	$f(x^*)$
Sphere	$f(x) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	$[0, 0, \dots, 0]$	0
Griewank	$f(x) = 1 + \sum_{i=1}^D \frac{1}{4000} x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]^D$	$[0, 0, \dots, 0]$	0
Rastrigin	$f(x) = 1 + \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	$[0, 0, \dots, 0]$	0
Ackley	$f(x) = 20 + e - 20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i))$	$[-32.768, 32.768]^D$	$[0, 0, \dots, 0]$	0
Rosenbrock	$f(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	$[1, 1, \dots, 1]$	0
Schwefel 2.22	$f(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	$[0, 0, \dots, 0]$	0

5.2 Optimal parameter selection

Population size NP and maximal iteration times $Cycle$ are essential parameters which have a significant impact on the performance of all population-based algorithms. The experimental results in [14,15,21] showed that the ACEDA obtains the best performance with large population sizes and few iteration times, while the ABC algorithm requires small population sizes and more iteration times [4–9]. As a result, experiments are designed in this part to select the best parameters for the novel hybrid algorithms Clayton ACABC and Gumbel ACABC.

In this paper, the fitness evaluation times, namely the

product of population and iteration times of different algorithms, are equal to each other to evaluate the performance of different algorithms fairly.

The number of fitness evaluation is set as 200 000 times, and five different parameter combinations are selected to optimize six benchmark functions respectively. The other parameters used in the algorithms are as follows. The dimension of all functions is $D = 100$, for Clayton ACABC the parameter with $\theta = 0.1$, while for Gumbel ACABC with $\theta = 1$. In different test environments, each algorithm runs 10 times and computes the mean value as the final results. Results for Clayton ACABC and Gumbel ACABC are shown in Table 3 and Table 4 respectively.

Table 3 Optimization results of Clayton ACABC with different population sizes and iterations

	Sphere	Griewank	Rastrigin	Ackley	Rosenbrock	Schwefel 2.22
$NP=40$ $Cycle=5\ 000$	1.5746e-015	2.2653e-016	2.0864e-010	5.4179e-014	98.091 6	3.8680e-015
$NP=100$ $Cycle=2\ 000$	7.7257e-016	3.5696e-016	1.3514e-006	7.5235e-012	331.400 7	1.2012e-015
$NP=400$ $Cycle=500$	1.3585e-012	8.5076e-013	52.162 7	1.6762e-007	175.529 3	0.213 9
$NP=1\ 000$ $Cycle=200$	0.015 0	0.007 6	196.643 6	0.017 0	105.235 6	0.219 8
$NP=2\ 000$ $Cycle=100$	48.102 1	1.413 5	433.739 4	2.188 0	491.650 9	11.615 6

Table 4 Optimization results of Gumbel ACABC with different population sizes and iterations

	Sphere	Griewank	Rastrigin	Ackley	Rosenbrock	Schwefel 2.22
$NP=40$ $Cycle=5\ 000$	7.7736e-016	3.1205e-016	1.4893e-011	4.7073e-014	96.5968	2.7707e-015
$NP=100$ $Cycle=2\ 000$	5.5341e-015	1.1102e-16	8.1194e-006	1.8652e-012	400.988 1	9.8912e-015
$NP=400$ $Cycle=500$	1.3585e-012	2.0768e-011	45.536 2	8.7059e-007	97.265 9	0.311 0
$NP=1\ 000$ $Cycle=200$	0.123 7	0.066 2	231.869 0	0.062 6	98.118 0	111.081 0
$NP=2\ 000$ $Cycle=100$	143.963 2	2.285 1	457.519 4	3.154 6	578.669 5	1 480.266

As shown in Table 3, data in bold are the best optimization results of each function. For Griewank, Rastrigin, Ackley and Rosenbrock, the best optimization result is obtained when the parameters are $NP = 40$, $Cycle = 5\ 000$, especially for Rastrigin, the optimization result is much better than the others. While for Sphere and Schwefel 2.22, the best optimization results are obtained when $NP = 100$, $Cycle = 2\ 000$. However, we can also observe that the optimization results when $NP = 40$, $Cycle = 5\ 000$ are nearly the same with the best ones.

According to the results shown in Table 4, for Griewank, the best optimization result is obtained when $NP = 100$, $Cycle = 2\ 000$, and when $NP = 40$, $Cycle = 5\ 000$, the optimization result is very close to the best result. While for the other five functions, the best optimization results are all obtained when $NP = 40$, $Cycle = 5\ 000$.

Based on the results shown in Table 3 and Table 4, we can draw the conclusion that both the Clayton ACABC algorithm and the Gumbel ACABC algorithm can get the

best performance when $NP = 40$, $Cycle = 5\ 000$. Thus in the following experiments, the population size is 40 and the iteration times is 5 000 for the Clayton ACABC algorithm and the Gumbel ACABC algorithm.

5.3 Performance comparison of different algorithms

In this part, six algorithms including ABC, GABC, Clayton CEDA, Gumbel CEDA, Clayton ACABC and Gumbel ACABC are used to optimize six benchmark functions respectively, and the performance on search accuracy and the convergent speed of the six algorithms are compared.

The number of fitness evaluation is set as 200 000 times. In ABC, GABC, Clayton ACABC and Gumbel ACABC, $NP = 40$ and $Cycle = 5\ 000$, while for Clayton CEDA and Gumbel CEDA, $NP = 100$ and $Cycle = 2\ 000$. The other parameters are set as follows. $D = 100$, $limit = 50$, $\theta = 0.1$ in Clayton ACABC and $\theta = 1$ in Gumbel ACABC. Each algorithm run 30 times and the mean value is calculated as the final results. Experimental results are shown in Table 5.

Table 5 Simulation results achieved by different algorithms

	ABC		GABC		Clayton CEDA	
	Mean	SD	Mean	SD	Mean	SD
Sphere	7.950 4e-014	3.149 5e-013	1.004 0e-014	2.089 2e-014	2.018 5e-008	5.673 8e-005
Griewank	1.617 8e-010	2.218 5e-009	6.050 3e-011	1.760 6e-007	5.121 6e-008	2.174 1e-005
Rastrigin	0.487 6	1.519 3	3.289 8e-004	0.003 1	123.626 4	15.128 3
Ackley	9.690 0e-009	1.251 0e-008	1.404 2e-012	9.117 5e-013	4.672 0e-006	1.201 3e-005
Rosenbrock	2.554 0	6.935 2	15.793 8	95.560 9	100.262 3	6.232 5
Schwefel 2.22	6.889 4e-010	9.560 0e-010	1.151 3e-014	4.012 3e-015	9.523 6e-007	5.468 5e-006
	Gumbel CEDA		Clayton ACABC		Gumbel ACABC	
	Mean	SD	Mean	SD	Mean	SD
Sphere	1.744 1e-011	7.263 8e-013	1.214 7e-015	2.542 8e-015	5.687 9e-016	1.120 3e-015
Griewank	1.172 9e-011	3.265 3e-014	5.121 3e-016	5.121 5e-012	1.110 2e-016	7.891 6e-013
Rastrigin	331.949 1	37.151 3	2.699 9e-010	2.226 7e-006	1.141 9e-011	6.874 9e-007
Ackley	7.640 2e-007	5.123 2e-006	5.524 5e-014	4.523 3e-014	1.723 1e-014	7.452 2e-015
Rosenbrock	97.222 7	12.727 6	97.538 5	2.365 3	96.563 2	5.129 6
Schwefel 2.22	2.073 0e-005	3.527 4e-008	3.527 9e-015	2.968 3e-016	2.721 3e-015	6.935 6e-016

As the results shown in Table 5, all of the six algorithms can optimize successfully on Sphere, Griewank, Ackley and Schwefel 2.22, and both Clayton ACABC and Gumbel ACABC can obtain optimization results with higher accuracy compared with the other four algorithms. The optimization results on Rastrigin indicate that only Clayton ACABC and Gumbel ACABC can optimize successfully with high precision, while ABC, GABC, Clayton CEDA and Gumbel CEDA cannot optimize Rastrigin successfully. However, the results on Rosenbrock show that all algorithms cannot achieve satisfying optimization results, and ABC and GABC outperform the CEDA and ACABC algorithms.

According to the optimization results, we can obtain

the conclusion that both Clayton ACABC and Gumbel ACABC can optimize most of the test functions with high precision and outperform ABC, GABC, Clayton CEDA, and Gumbel CEDA.

In order to evaluate the convergent speed of the proposed ACABC algorithms, the convergence curves of five functions except Rosenbrock are drawn in Figs. 2–6 respectively. It can be observed that Clayton ACABC and Gumbel ACABC accelerate the search speed and improve the convergent speed significantly.

According to the experimental results in this part, we can draw the conclusion that Clayton ACABC and Gumbel ACABC can speed up the convergent speed with greater precision for most test functions. The novel hybrid

ACABC algorithms outperform the ABC algorithms and ACEDA.

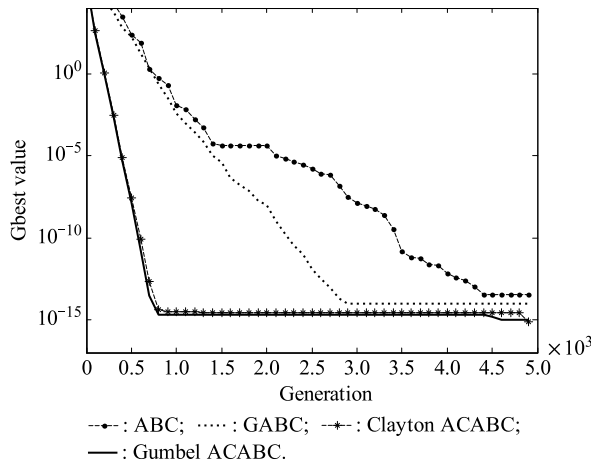


Fig. 2 Convergence curve on Sphere (Sphere with dim = 100)

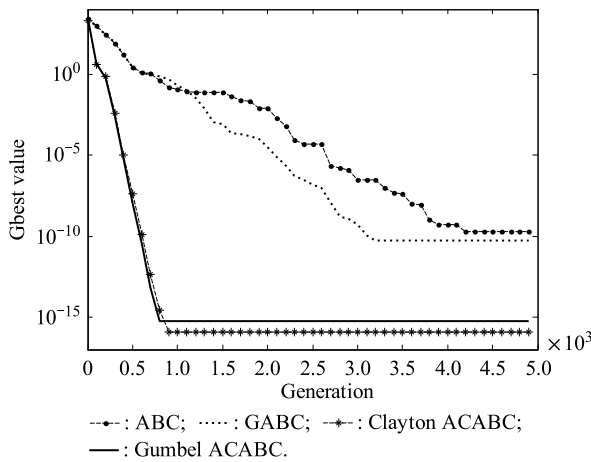


Fig. 3 Convergence curve on Griewank (Griewank with dim = 100)

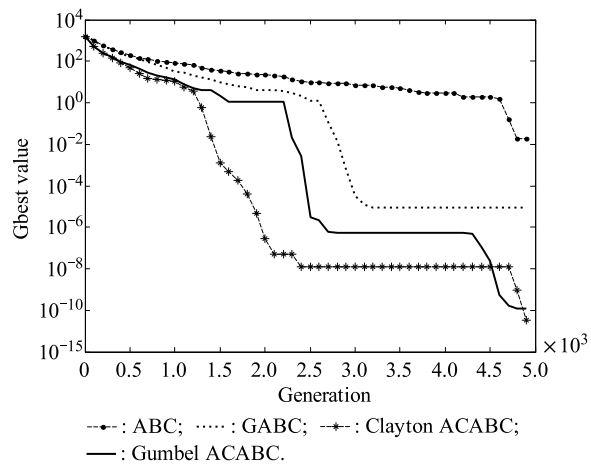


Fig. 4 Convergence curve on Rastrigin (Rastrigin with dim = 100)

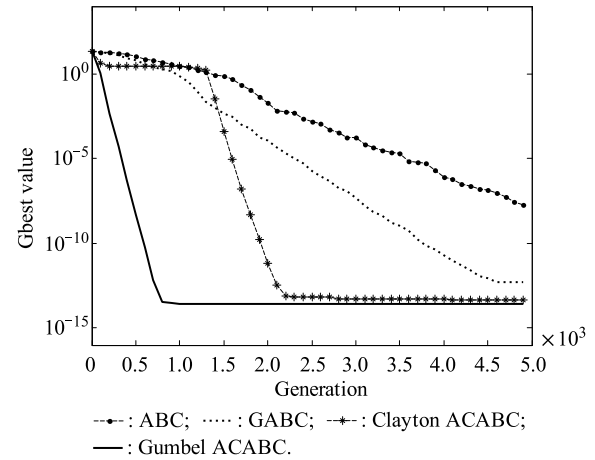


Fig. 5 Convergence curve on Ackley (Ackley with dim = 100)

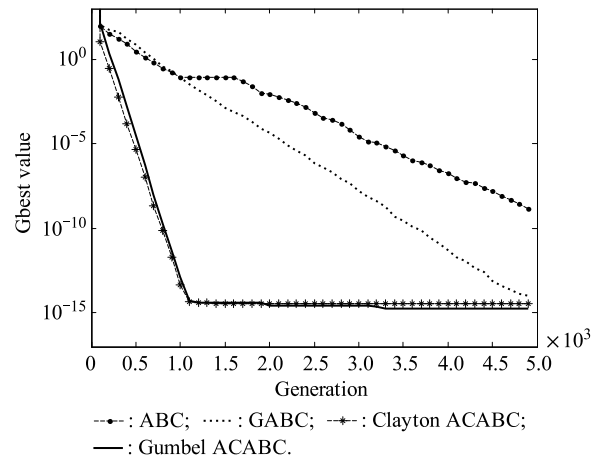


Fig. 6 Convergence curve on Schwefel 2.22 (Schwefel 2.22 with dim = 100)

6. Conclusions

In this paper, we propose a novel hybrid algorithm called ACABC algorithm. The ABC algorithm is a biological-inspired optimization algorithm with a random search process. In order to improve the search efficiency, we introduce ACEDA into the onlooker stage and propose a more intelligent onlooker mechanism. In the modified mechanism, onlookers first sort all solutions obtained by employed bees in descending order according to the fitness values, select excellent individuals to construct dominant population, then construct a probability distribution model based on ACEDA, and sample based on the estimated distribution to generate new individuals. The updating strategy ensures that the new population obeys the distribution of the dominant population. In addition, we modify the neighborhood search strategy with the gbest value by guiding the operator and the adaptive parameter to improve the exploitation of the algorithm. Experimental results show that the ACABC algorithm speeds up the convergent speed

with greater precision for most test functions. In conclusion, the novel hybrid ACABC algorithms outperform the ABC algorithms and ACEDA.

References

- [1] K. S. Tang, K. F. Man, S. Kwong, et al. Genetic algorithms and their applications. *IEEE Signal Processing Magazine*, 1996, 13(6): 22–37.
- [2] S. Baluja. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Pittsburgh: Carnegie Mellon University, 1994.
- [3] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical report-tr06. Kayseri: Erciyes University, 2005.
- [4] D. Karaboga, B. Basturk. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 2008, 8(1): 687–697.
- [5] D. Karaboga, B. Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 2009, 214(1): 108–132.
- [6] G. Zhu, S. Kwong. Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 2010, 217(7): 3166–3173.
- [7] X. Lei, X. Huang, A. Zhang. Improved artificial bee colony algorithm and its application in data clustering. *Proc. of the 5th IEEE International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, 2010: 514–521.
- [8] B. Akay, D. Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 2012, 192(1): 120–142.
- [9] W. Gao, S. Liu. Improved artificial bee colony algorithm for global optimization. *Information Processing Letters*, 2011, 111(17): 871–882.
- [10] H. Muhlenbein, G. PaaB. From combination of genes to the estimation of distribution: binary parameters. *Proc. of the Conference on Lecture Notes in Computer Science 1411, Parallel Problem Solving from Nature-PPSN IV, H. M. Voigt(eds.)*. 1996: 178–187.
- [11] S. Baluja. Population-based incremental learning: a method for integrating genetic search based on function optimization and competitive learning. Pittsburgh: Carnegie Mellon University, 1994.
- [12] M. Pelikan, D. E. Goldberg, E. Cantu-paz. BOA: the Bayesian optimization algorithm. *Proc. of the Conference on the Genetic and Evolutionary Computation*, 1999: 525–532.
- [13] F. Durante, C. Sempì. Copula theory: an introduction. *Copula Theory and Its Applications*, 2010(198): 3–31.
- [14] L. F. Wang, J. C. Zeng, Y. Hong. Estimation of distribution algorithm based on archimedean copulas. *Proc. of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, 2009: 993–996.
- [15] Y. Gao. Multivariate estimation of distribution algorithm with laplace transform archimedean copula. *Proc. of the International Conference on Information Engineering and Computer Science*, 2009: 1–5.
- [16] R. B. Nelsen. *An introduction to copulas*. 2nd ed. New York: Springer, 2006.
- [17] D. Y. Li, L. Peng. Goodness-of-fit test for tail copulas modeled by elliptical copulas. *Statistics and Probability Letters*, 2009, 79(8): 1097–1104.
- [18] M. Denuit, O. Purcaruff, I. Van Keilegorni. Bivariate Archimedean copula models for censored data in non-life insurance. Brussel: Université catholique de Louvain, 2006.
- [19] M. R. F. Mendez, R. Landa. An EDA based on Bayesian networks constructed with Archimedean copulas. *Proc. of the 4th World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 2012: 188–193.
- [20] A. W. Marshall, I. Olkin. Families of multivariate distributions. *Journal of the American Statistical Association*, 1988, 83(403): 834–841.
- [21] L. F. Wang. The research on estimation of distribution algorithm based on copula theory. Lanzhou: Lanzhou University of Technology, 2011: 37–50.

Biographies



Haidong Xu was born in 1990. She received her B.S. degree from Shandong University in June, 2012 in communication engineering and now she is an M.Sc. student in the same University. Her main research area includes artificial intelligence and computing intelligence, artificial neural network, swarm intelligence algorithms and mathematical statistics.
E-mail: xu-hai-dong1990@163.com



Mingyan Jiang was born in 1964. He received his M.S. degree from Shandong University in 1992 and his Ph.D. degree in 2005. He finished his postdoctoral research in Spain (CTTC) in communication signal and system in 2007. Now he is a full professor and a doctoral supervisor in the School of Information Science and Engineering in Shandong University, China. His research interests include soft computing, signal and image processing, computer network, artificial intelligence and data mining. He has published more than 200 professional papers and 6 books.
E-mail: jiangmingyan@sdu.edu.cn



Kun Xu was born in 1988. He received his B.S. degree from Shandong Normal University in June, 2011 in electronic information engineering and now is an M.Sc. student in Shandong University. His main research interests include machine learning, parallel computation, optimization and artificial intelligence.
E-mail: xukun_sdu@163.com