

Research Article

A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm

Wenping Zou,^{1,2} Yunlong Zhu,¹ Hanning Chen,¹ and Xin Sui^{1,2}

¹ Key Laboratory of Industrial Informatics, Shenyang Institute of Automation,
Chinese Academy of Sciences, Shenyang 110016, China

² Graduate School of the Chinese Academy of Sciences, Beijing 100039, China

Correspondence should be addressed to Wenping Zou, zouwp@sia.cn

Received 25 June 2010; Revised 23 September 2010; Accepted 15 October 2010

Academic Editor: Daniel Czumanski

Copyright © 2010 Wenping Zou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Artificial Bee Colony (ABC) is one of the most recently introduced algorithms based on the intelligent foraging behavior of a honey bee swarm. This paper presents an extended ABC algorithm, namely, the Cooperative Article Bee Colony (CABC), which significantly improves the original ABC in solving complex optimization problems. Clustering is a popular data analysis and data mining technique; therefore, the CABC could be used for solving clustering problems. In this work, first the CABC algorithm is used for optimizing six widely used benchmark functions and the comparative results produced by ABC, Particle Swarm Optimization (PSO), and its cooperative version (CPSO) are studied. Second, the CABC algorithm is used for data clustering on several benchmark data sets. The performance of CABC algorithm is compared with PSO, CPSO, and ABC algorithms on clustering problems. The simulation results show that the proposed CABC outperforms the other three algorithms in terms of accuracy, robustness, and convergence speed.

1. Introduction

Swarm Intelligence (SI) is an innovative artificial intelligence technique for solving complex optimization problems. In recently years, many SI algorithms have been proposed, such as Ant Colony Optimization (ACO) [1], Particle Swarm Algorithm (PSO) [2], and Bacterial Foraging Optimization (BFO) [3]. Artificial Bee Colony (ABC) algorithm is a new swarm intelligent algorithm that was first introduced by Karaboga in Erciyes University of Turkey in 2005 [4], and the performance of ABC is analyzed in 2007 [5]. The ABC algorithm imitates the behaviors of real bees in finding food sources and sharing the information with other bees. Since ABC algorithm is simple in concept, easy to implement, and has fewer control parameters, it has been widely used in many fields. Until now, ABC has been applied successfully to some engineering problems, such as constrained optimization problems [6], neural networks [7], and clustering [8].

However, like other stochastic optimization algorithms, such as PSO and Genetic Algorithm (GA), as the dimensionality of the search space increases, ABC algorithm possesses a poor convergence behavior. Cooperative search is one of the solutions to this problem, which has been extensively studied in the past decade. Potter proposed cooperative coevolutionary genetic algorithm (CCGA) [9], Van den Bergh and Engelbrecht proposed cooperative particle swarm optimizer, called CPSO [10], and Chen et al. proposed cooperative bacterial foraging optimization [11]. This paper applies Potter's cooperative search technique to the ABC, resulting in a new cooperative ABC algorithm, namely CABC. In order to evaluate the performance of the CABC, we compared the performance of the CABC algorithm with that of ABC, PSO, and CPSO on a set of well-known benchmark functions such as Rosenbrock, Griewank, Rastrigin, Ackley, and Schwefel. From the simulation results, the CABC algorithm shows remarked performance improvement over the other algorithms in all benchmark functions.

Data clustering is the process of grouping data into a number of clusters. The goal of data clustering is to make the data in the same cluster share a high degree of similarity while being very dissimilar to data from other clusters. Clustering algorithms have been applied to a wide range of problems, such as data mining [12], data analysis, pattern recognition [13], and image segmentation [14]. Clustering algorithms can be simply classified as hierarchical clustering and partitional clustering [15]. This paper mainly focuses on partitional clustering. Partitional clustering algorithm divides data vectors into a predefined number of clusters by optimizing some certain criterion. The most popular partitional clustering algorithm is K-means. In the past three decades, K-means clustering algorithm has been used in various domains. However, K-means algorithm is sensitive to the initial states and always converges to the local optimum solution. In order to overcome this problem, many methods have been proposed, such as Zhang and Hsu have introduced K-harmonic means (KHM) [16], and Bezdek has proposed fuzzy c-means (FCM) clustering [17]. Over the last decade, more and more stochastic, population-based optimization algorithms have been applied to clustering problems. For instance, Shelokar et al. have introduced an evolutionary algorithm based on ACO algorithm for clustering problem [18, 19], Merwe et al. have presented PSO to solve the clustering problem [20, 21], and Karaboga and Ozturk have used the ABC algorithm [22]. In this paper, a CABC algorithm is applied to solve the clustering problem, which has been tested on a variety of data sets. The performance of the CABC on clustering is compared with results of the ABC, PSO, and CPSO algorithms on the same data sets. The above data sets are provided from the UCI database [23].

The rest of the paper is organized as follows. In Section 2 we will introduce the original ABC algorithm. Section 3 will discuss cooperative search methods, and our cooperative implementations of the ABC algorithm will be presented. The details of CABC algorithm will be given in this section. Section 4 tests the algorithms on the benchmarks, and the results obtained are presented and discussed. In Section 5, the cluster analysis problem is discussed. The application of CABC algorithm on clustering is shown in Section 6, and the performance of CABC algorithm is compared with PSO, CPSO, and ABC algorithms on clustering problem in this section. Finally, conclusions are given in Section 7.

2. The Original ABC Algorithm

The artificial bee colony algorithm is a new population-based metaheuristic approach, initially proposed by Karaboga [4, 5] and further developed by Karaboga and Basturk

[6, 7]. It has been used in various complex problems. The algorithm simulates the intelligent foraging behavior of honey bee swarms. The algorithm is very simple and robust. In the ABC algorithm, the colony of artificial bees is classified into three categories: employed bees, onlookers, and scouts. Employed bees are associated with a particular food source that they are currently exploiting or are “employed” at. They carry with them information about this particular source and share the information to onlookers. Onlooker bees are those bees that are waiting on the dance area in the hive for the information to be shared by the employed bees about their food sources, and then make decision to choose a food source. A bee carrying out random search is called a scout. In the ABC algorithm, the first half of the colony consists of the employed artificial bees and the second half includes the onlookers. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source has been exhausted by the bees becomes a scout. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution represented by that food source. Onlookers are placed on the food sources by using a probability-based selection process. As the nectar amount of a food source increases, the probability value with which the food source is preferred by onlookers increases, too [4, 5]. The main steps of the algorithm are given below.

In the initialization phase, the ABC algorithm generates a randomly distributed initial food source positions of SN solutions, where SN denotes the size of employed bees or onlooker bees. Each solution x^i ($i = 1, 2, \dots, SN$) is a D -dimensional vector. Here, D is the number of optimization parameters. And then evaluate each nectar amount fit_i . In the ABC algorithm, nectar amount is the value of benchmark function.

In the employed bees’ phase, each employed bee finds a new food source v_i in the neighborhood of its current source x_i . The new food source is calculated using the following expression:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \quad (2.1)$$

where $k \in (1, 2, \dots, SN)$ and $j \in (1, 2, \dots, D)$ are randomly chosen indexes, and $k \neq i$. ϕ_{ij} is a random number between $[-1, 1]$. And then employed bee compares the new one against the current solution and memorizes the better one by means of a greedy selection mechanism.

In the onlooker bees’ phase, each onlooker chooses a food source with a probability which is related to the nectar amount (fitness) of a food source shared by employed bees. Probability is calculated using the following expression:

$$p^i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}. \quad (2.2)$$

In the scout bee phase, if a food source can not be improved through a predetermined cycles, called “limit”, it is removed from the population, and the employed bee of that food source becomes scout. The scout bee finds a new random food source position using the

Main steps of the ABC algorithm.

- (2.1) cycle=1
- (2.2) Initialize the food source positions $x^i, i = 1, \dots, SN$
- (2.3) Evaluate the nectar amount (fitness function fit_i) of food sources
- (4) **repeat**
- (5) Employed Bees' Phase
 - For each employed bee
 - Produce new food source positions v_i
 - Calculate the value fit_i
 - Apply greedy selection mechanism
 - EndFor.
- (6) Calculate the probability values p_i for the solution.
- (7) Onlooker Bees' Phase
 - For each onlooker bee
 - Chooses a food source depending on p_i
 - Produce new food source positions v_i
 - Calculate the value fit_i
 - Apply greedy selection mechanism
 - EndFor
- (8) Scout Bee Phase
 - If there is an employed bee becomes scout
 - Then replace it with a new random source positions
- (9) Memorize the best solution achieved so far
- (10) cycle=cycle+1.
- (11) **until** cycle=Maximum Cycle Number

Algorithm 1: Pseudocode for ABC algorithm.

equation below:

$$x_i^j = x_{\min}^j + \text{rand}[0, 1] \left(x_{\max}^j - x_{\min}^j \right), \quad (2.3)$$

where x_{\min}^j and x_{\max}^j are lower and upper bounds of parameter j , respectively.

These steps are repeated through a predetermined number of cycles, called Maximum Cycle Number (MCN), or until a termination criterion is satisfied [4, 5, 24].

3. Cooperative ABC

In the ABC algorithm, the goal of each individual bee is to produce the best solution. From expression (2.1), we can see that the new food source is produced by a random neighborhood of current food position and a random single dimension of D -dimensional vector. This will bring about a problem that an individual may have discovered a good dimension, but the fitness of the individual is computed by using D -dimensional vector, hence we know it is very probable that the individual is not the best solution in the end, and the good dimension which the individual has found will be abandoned. For example, there are two vectors $x_1(10, 8)$ and $x_2(8, 10)$, and the fitness function $f(x) = \|x\|^2$, where the global minimizer of the fitness function is $(0, 0)$. Set x_1 and x_2 as the current positions. And $x'_1(8, 6)$ and $x'_2(4, 12)$ represent the final positions which x_1 and x_2 can find. We can see that $f(x'_1) = 100$ is smaller than

$f(x'_2) = 160$, so we select x'_1 as the solution of the function. However, we notice that the first dimension of x'_2 is 4 which is closer to the global minimizer than the first dimension of x'_1 . Because x'_2 is not the best solution, the first dimension of x'_2 is not selected. If we can save the good dimension, the final solution will become $x''_1(4, 6)$ and the quality of solution will be significantly improved. Therefore, we hope to find every best dimension in all individuals. We need each individual's contribution to the best solution.

The same problem is met in the GA and PSO algorithms. Cooperative search technique has been applied to GA and PSO [9, 10]. To produce a good solution vector, all the populations must cooperate. And the information from all the populations need to be used. Therefore, we apply cooperative search to solve the problem in the ABC algorithm and propose the Cooperative ABC algorithm. In the CABC algorithm, we set a super best solution vector, namely, g_{best} and its each component of D -dimensional is the best in all populations. For $g_{best}: (g_1, g_2, \dots, g_i, \dots, g_D)$ g_i corresponds to the i th component of the g_{best} . The fitness function is represented by f . in algorithm 1 The main steps of CABC algorithm are given.

In the initialization phase, we evaluate the fitness of the initial food source positions and set the position which has the best fitness as the initial g_{best} .

In the employed bees' and onlooker bees' phase, we use the j component of each individual to replace the corresponding component of the g_{best} to find the best position of the j component. G_{best} do not influence the employed and onlooker bees finding new food sources. It is a virtual bee. It just saves the best one of each component. After all phases, the best solution achieved by all individuals and the g_{best} will be compared.

4. Experiments

4.1. Benchmark Functions

In order to compare the performance of the proposed CABC algorithm with ABC, PSO, and CPSO, we used six well-known benchmark functions. One of the benchmark functions is unimodal and the others have a number of local minima [25].

The first function is Sphere function and its global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-5.12, 5.12]$. It is a unimodal function and easy to solve. The Sphere function is defined as

$$f_1(x) = \sum_{i=1}^D x_i^2. \quad (4.1)$$

The second function is Rosenbrock function and its global minimum value is 0 at $(1, 1, \dots, 1)$. Initialization range for the function is $[-15, 15]$. It is a multimodal function. It has a narrow valley near the global optimum and it is difficult to converge the global optimum. The Rosenbrock function is defined as

$$f_2(x) = \sum_{i=1}^D 100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2. \quad (4.2)$$

The third function is Griewank function and its global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-600, 600]$. This function has a product

Main steps of the ABC algorithm

- (2.1) cycle=1
- (2.2) Initialize the food source positions $x_i, i = 1 \cdot SN$
- (2.3) Evaluate the nectar amount (fitness fit_i) of food sources and find the best food source which is the initial value of g_{best}
- (4) **repeat**
- (5) For each component $j \in (1, 2, \dots, D)$
- (6) Employed Bees' Phase
 - For each employed bee $i = 1 \dots SN$
 - Replace the j component of the g_{best} by using the j component of bee i
 - Calculate the $f[newg_{best}(g_1, g_2, \dots, x_{ij}, \dots, g_D)]$
 - If $f(newg_{best})$ better than $f(g_{best})$
 - Then g_{best} is replaced by $newg_{best}$
 - For employed bee i produce new food source positions v_i by using (2.1)
 - Calculate the value fit_i
 - Apply greedy selection mechanism
 - EndFor.
- (7) Calculate the probability values p_i for the solution.
- (8) Onlooker Bees' Phase
 - For each onlooker bee $i = 1 \dots SN$
 - Chooses a food source depending on p_i
 - Replace the j component of the g_{best} by using the j component of bee i
 - Calculate the $f[newg_{best}(g_1, g_2, \dots, x_{ij}, \dots, g_D)]$
 - If $f(newg_{best})$ better than $f(g_{best})$
 - Then g_{best} is replaced by $newg_{best}$
 - For onlooker bee i produce new food source positions v_i by using (2.1)
 - Calculate the value fit_i
 - Apply greedy selection mechanism
 - EndFor
- EndFor
- (9) Scout Bees' Phase
 - If there is an employed bee becomes scout
 - Then replace it with a new random source positions
- (10) Memorize the best solution achieved so far
- (11) Compare the best solution with g_{best} and Memorize the better one.
- (12) cycle=cycle+1.
- (13) **until** cycle=Maximum Cycle Number

Algorithm 2: Pseudocode for ABC algorithm.

term, introducing an interdependency between the variables, thereby making it difficult to reach the global optimum. The Griewank function is defined as

$$f_3(x) = \frac{1}{4000} \left(\sum_{i=1}^D x_i^2 \right) - \left(\prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) \right) + 1. \quad (4.3)$$

The fourth function is Rastrigin function and its global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-15, 15]$. It is a complex multimodal

function, and it has a large number of local optima. Thereby it is easy for optimization algorithm to trap in local optimum. The Rastrigin function is defined as

$$f_4(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10). \quad (4.4)$$

The fifth function is Ackley function and its global minimum value is 0 at $(0, 0, \dots, 0)$. Initialization range for the function is $[-32.768, 32.768]$. This function has one narrow global optimum. The second exponential term that covers its surface with many local minima. The Ackley function is defined as

$$f_5(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right). \quad (4.5)$$

The sixth function is Schwefel function and its global minimum value is 0 at $(420.9867, 420.9867, \dots, 420.9867)$. Initialization range for the function is $[-500, 500]$. The Schwefel function is very complex. There are a lot of peaks and valleys on its surface. There is a second best minimum far from the global minimum and many algorithms fall into it:

$$f_6(x) = D * 418.9829 + \sum_{i=1}^D -x_i \sin\left(\sqrt{|x_i|}\right). \quad (4.6)$$

4.2. Parameter Settings for the Involved Algorithms

In the experiment all functions are tested on thirty dimensions, and the population size of all algorithms was 100. The PSO algorithm we used is the standard PSO. In PSO and CPSO algorithm, inertia weight ω varies from 0.9 to 0.7 linearly with the iterations and the acceleration factors c_1 and c_2 being both 2.0 [26]. In order to compare the different algorithms, a fair time measure must be selected. It was, therefore, decided to use the number of function evaluations (FEs) as a time measure [10]. Thereby FEs in a time measure is our termination criterion.

4.3. Simulation Results for Benchmark Functions

The experimental results, including the best, worst, average, and standard deviation of the function values found in 30 runs are proposed in Table 1 and all algorithms were terminated after 100,000 function evaluations.

From Table 1, the CAB algorithm is better than the other algorithms on four benchmark functions (Sphere, Rastrigin, Ackley, Schwefel) while the ABC algorithm shows better performance than the other algorithms on two benchmark functions (Rosenbrock, Griewank). The PSO converges very slowly and its performance is very bad on four benchmark functions (Griewank, Rastrigin, Ackley, Schwefel), as can be seen in Figure 1.

On Sphere function, all algorithms perform very well. However, Table 1 shows that the performance of CAB is much better than the others'. The speed of convergence of CAB is much faster, as can be seen in Figure 1(a).

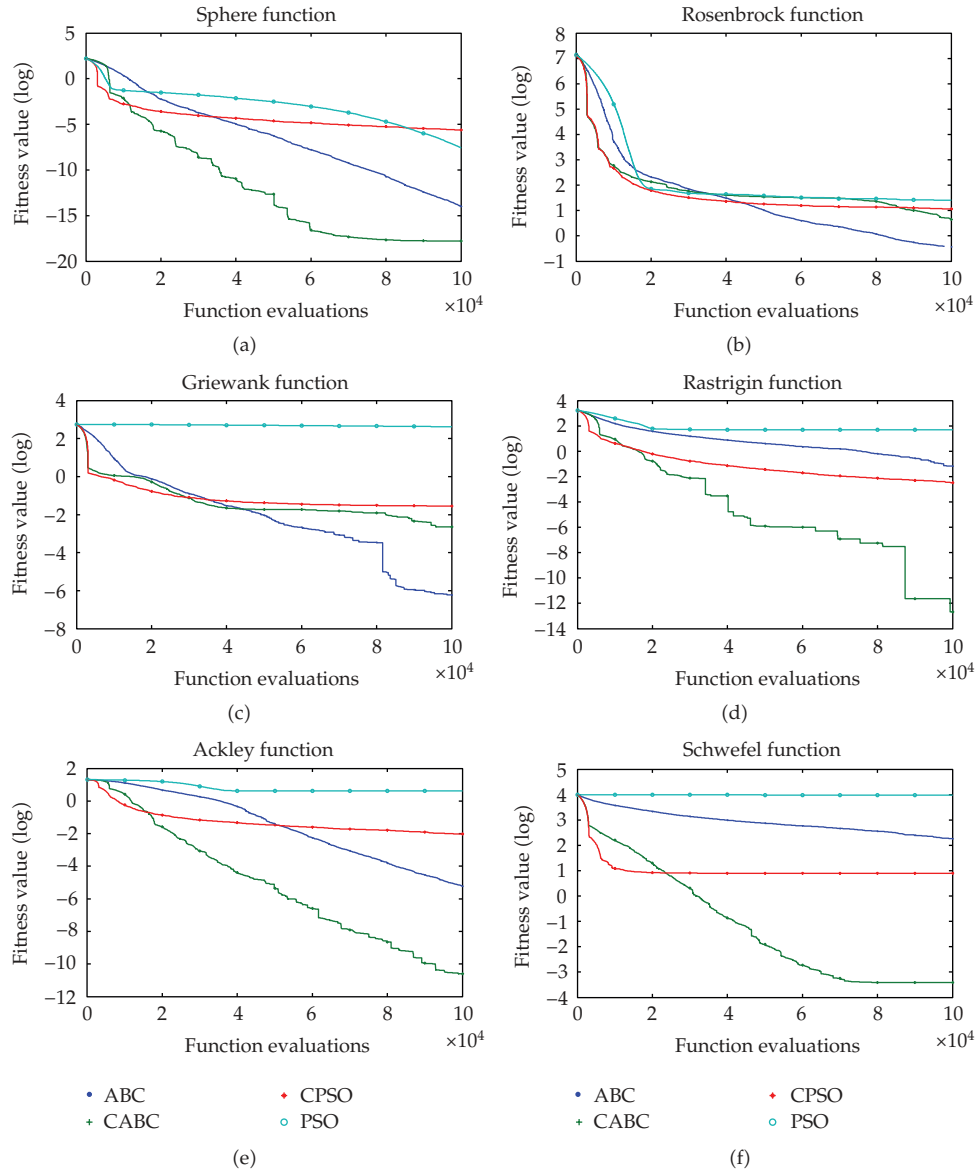


Figure 1: The median convergence characteristics of fitness functions. (a) Sphere function. (b) Rosenbrock function. (c) Griewank function. (d) Rastrigin function. (e) Ackley function. (f) Schwefel function.

On Rosenbrock function, the ABC algorithm shows better performance in terms of values of average, best, and standard deviation than the other methods. The CABC algorithm is a little worse than the ABC. We can see that the CABC, PSO, and CPSO algorithm converge fast at first, but they become trapped in a local minimum very soon, as can be seen in Figure 1(b).

On Griewank function, the CABC algorithm performs much worse than ABC, but it is better than CPSO and PSO algorithms. Table 1 shows that the best value of the fitness function for CABC algorithm is 0 much smaller than this of the other four methods. Nevertheless,

Table 1: Results comparison of different optimal algorithms for 30 runs.

	30D	ABC	CABC	PSO	CPSO
Sphere	Average	1.1396e -014	1.5046e-018	2.8575e -008	2.1792e -006
	Best	2.1069e -015	5.9342e -019	1.6427e -009	4.8957e -007
	Worst	3.4078e -014	2.7426e -018	2.1292e -007	5.7575e -006
	Std	8.0826e -015	5.4311e-019	3.8261e -008	1.4044e -006
Rosenbrock	Average	3.3325e - 001	7.1667e + 000	2.6555e + 001	1.4528e + 001
	Best	1.3804e - 002	2.3045e - 002	7.8515e + 000	1.3222e - 003
	Worst	1.1599e + 000	7.5039e + 001	9.6353e + 001	7.7598e + 001
	Std	2.3784e - 001	1.3472e + 001	1.7080e + 001	2.7135e + 001
Griewank	Average	6.0208e - 007	3.9523e - 003	4.1956e + 002	3.2394e - 002
	Best	1.3549e - 011	0	3.6532e + 002	3.3059e - 007
	Worst	1.7767e - 005	1.2665e - 001	4.7479e + 002	1.5611e - 001
	Std	3.2419e - 006	2.7866e - 002	2.9238e + 001	4.1144e - 002
Rastrigin	Average	1.8603e - 001	1.3642e - 013	4.6671e + 001	3.0123e - 003
	Best	3.8257e - 009	0	2.1889e + 001	7.4135e - 004
	Worst	9.9496e - 001	3.4106e - 013	8.2581e + 001	6.0268e - 003
	Std	3.6710e - 001	8.1216e - 014	1.2656e + 001	1.5145e - 003
Ackley	Average	6.0643e - 006	8.4976e - 012	4.2520e + 000	9.0169e - 003
	Best	1.5905e - 006	2.5553e - 012	2.8872e + 000	4.5035e - 003
	Worst	1.7602e - 005	2.9208e - 011	5.7625e + 000	1.4871e - 002
	Std	3.5254e - 006	7.2161e - 012	8.3370e - 001	2.6789e - 003
Schwefel	Average	1.9897e + 002	3.8183e - 004	9.5252e + 003	7.8976e + 000
	Best	4.1238e - 004	3.8183e - 004	8.7622e + 003	7.2405e - 004
	Worst	4.7389e + 002	3.8186e - 004	1.0151e + 004	1.1844e + 002
	Std	1.1697e + 002	6.3175e - 009	3.7111e + 002	3.0049e + 001

CABC algorithm is easily trapped at local optimum, so the average value is worse than ABC. In the initialization phase, the CABC and CPSO algorithms perform well, but their performance rapidly deteriorates when they run about 30,000 function evaluations, as can be seen in Figure 1(c).

From the results of Rastrigin and Ackley, we can observe that the ability of exploiting the optimum of the CABC algorithm is very strong. The CABC is definitely better than the other four methods for these two test functions. On Rastrigin function, the CPSO is more successful in terms of all values than ABC. However, on Ackley function, ABC algorithm outperforms CPSO. We can see that PSO algorithm is easily trapped in local optimum on Rastrigin and Ackley function. From Figures 1(d) and 1(e), we can observe that the CABC algorithm is able to continue improving its solution on these two functions.

On Schwefel function, it is a very complex function and there is a second best minimum that many algorithms fall into. The performance of ABC and PSO algorithms deteriorate in optimizing this function and the CPSO algorithm performs a little better than them. However, the CABC algorithm shows better performance on this, as can be seen in Figure 1(f). From Table 1, we notice that the average, best, worst, and standard deviation values of the CABC algorithm are almost the same. That means that CABC falls into this position every time.

From the results above, it is concluded that the proposed algorithm can be efficiently used for multivariable, multimodal function optimization. For that reason, in this paper, we apply our CAB algorithm to solve clustering problem.

5. Data Clustering

5.1. *K*-Means Algorithm

As mentioned above, the goal of data clustering is grouping data into a number of clusters and *K*-means algorithm is the most popular clustering algorithm. In this section, we briefly describe the *K*-means algorithm. Let $X = (x_1, x_2, \dots, x_n)$ be a set of n data and let each data vector be a p -dimensional vector. Let $C = \{c_1, c_2, \dots, c_k\}$ be a set of K clusters and K denotes the number of cluster centroids which is provided by the user. In *K*-means algorithm, firstly, randomly initialize the K cluster centroid vectors and then assign each data vector to the class with the closest centroid vector. In this study, we will use Euclidian metric as a distance metric. The expression is given as follows:

$$d(x_i, c_j) = \sqrt{\sum_{k=1}^p (x_{ik} - c_{jk})^2}. \quad (5.1)$$

After all data being grouped, recalculate the cluster centroid vectors using

$$c_j = \frac{1}{n_j} \sum_{x_i \in c_j} x_i, \quad (5.2)$$

where n_j is the number of data vectors which belong to cluster j . After the above process, reassign the data to the new cluster centroids and repeat the process until a criterion is satisfied. In this study, the criterion is when the maximum number of iterations has been exceeded. To know whether the partition is good or not, a measure for partition must be defined. A popular performance function for measuring goodness of the partition is the total within-cluster variance or the total mean-square quantization error (MSE) [27, 28], which is defined as follows:

$$\text{Perf}(X, C) = \sum_{i=1}^N \text{Min} \left\{ \|X_i - C_l\|^2 \mid l = 1, \dots, K \right\}. \quad (5.3)$$

Because *K*-means algorithm is sensitive to the initial states and always converges to the local optimum solution, more population-based stochastic search algorithms are presented. In this paper, we will use CAB algorithm to solve clustering problem.

5.2. CAB Algorithm on Clustering

In the CAB algorithm, each individual represents a solution in K dimensional space. The number of dimension is equal to the number of clusters. Each component of an individual

```

For data vector  $x_i$ 
  Calculate the Euclidean distance by using (5.1)
  Assign  $x_i$  to the closest centroid cluster  $c_j$ .
  Calculate the measure function using equation (5.3)
EndFor.
Return value of the fitness function.

```

Algorithm 3: Main steps of the fitness function.

represents a cluster centroid and each cluster centroid is a p -dimensional vector. In the initialization phase, we use maximum and minimum value of each component of the data set (which is to be grouped) as CABC algorithm individuals' initialization range. And initial solution is randomly generated in this range. We use expression (5.3) to calculate the fitness function of individuals. Here the main steps of the fitness function are given in algorithm 3

6. Data Clustering Experimental Results

To evaluate performance of the proposed CABC approach for clustering, we compare the results of the K -means, PSO, CPSO, ABC, and CABC clustering algorithms using six different data sets which are selected from the UCI machine learning repository [23].

Motorcycle data ($N = 133$, $d = 2$, $K = 4$): the Motorcycle benchmark consists of a sequence of accelerometer readings through time following a simulated motorcycle crash during an experiment to determine the efficacy of crash helmets.

Iris data ($N = 150$, $d = 4$, $K = 3$): this data set with 150 random samples of flowers from the iris species *setosa*, *versicolor*, and *virginica* collected by Anderson (1935). From each species there are 50 observations for sepal length, sepal width, petal length, and petal width in cm. This data set was used by Fisher (1936) in his initiation of the linear-discriminant-function technique [29].

Wine data ($N = 178$, $d = 13$, $K = 3$): this is the wine data set, which is also taken from MCI laboratory. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There are 178 instances with 13 numeric attributes in wine data set. All attributes are continuous. There is no missing attribute value [29].

Contraceptive Method Choice ($N = 1473$, $d = 10$, $K = 3$): this data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics [29].

Wisconsin breast cancer ($N = 683$, $d = 9$, $K = 2$), which consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects) [29].

Table 2: Comparison of intracluster distances for the five clustering algorithms.

Data set	Criteria	CABC	ABC	PSO	CPSO	K-means
Motor cycle	Average	$2.0607e + 003$	$2.0689e + 003$	$2.0758e + 003$	$2.0690e + 003$	$3.0123e + 003$
	Best	$2.0606e + 003$	$2.0606e + 003$	$2.0606e + 003$	$2.0606e + 003$	$2.4463e + 003$
	Worst	$2.0624e + 003$	$2.1267e + 003$	$2.2125e + 003$	$2.1103e + 003$	$4.6832e + 003$
	Std	$3.2158e - 001$	$1.9118e + 001$	$4.6354e + 001$	$1.3340e + 001$	$4.3906e + 002$
Iris	Average	$9.4603e + 001$	$9.4607e + 001$	$9.7526e + 001$	$9.5614e + 001$	$1.0605e + 002$
	Best	$9.4603e + 001$	$9.4603e + 001$	$9.4898e + 001$	$9.4603e + 001$	$9.7333e + 001$
	Worst	$9.4603e + 001$	$9.4644e + 001$	$1.1914e + 002$	$1.0493e + 002$	$1.2045e + 002$
	Std	$1.9767e - 009$	$7.7734e - 003$	$4.4576e + 000$	$1.9581e + 000$	$1.4631e + 001$
Wine	Average	$1.6294e + 004$	$1.6298e + 004$	$1.6372e + 004$	$1.6302e + 004$	$1.8061e + 004$
	Best	$1.6292e + 004$	$1.6294e + 004$	$1.6294e + 004$	$1.6292e + 004$	$1.6555e + 004$
	Worst	$1.6296e + 004$	$1.6302e + 004$	$1.6723e + 004$	$1.6384e + 004$	$1.8563e + 004$
	Std	$1.5466e + 001$	$6.2411e + 000$	$1.0718e + 002$	$1.8266e + 001$	$7.9321e + 002$
CMC	Average	$5.6938e + 003$	$5.6954e + 003$	$5.7293e + 003$	$5.6974e + 003$	$5.8936e + 003$
	Best	$5.6937e + 003$	$5.6939e + 003$	$5.6942e + 003$	$5.6938e + 003$	$5.8422e + 003$
	Worst	$5.6939e + 003$	$5.6986e + 003$	$5.8777e + 003$	$5.7107e + 003$	$5.9344e + 003$
	Std	$4.5501e - 002$	$1.3824e + 000$	$4.0245e + 001$	$4.0160e + 000$	$4.7165e + 001$
Cancer	Average	$2.9644e + 003$	$2.9644e + 003$	$2.9656e + 003$	$2.9650e + 003$	$3.2512e + 003$
	Best	$2.9644e + 003$	$2.9644e + 003$	$2.9644e + 003$	$2.9644e + 003$	$2.9991e + 003$
	Worst	$2.9644e + 003$	$2.9644e + 003$	$2.9726e + 003$	$2.9694e + 003$	$3.5215e + 003$
	Std	$1.8380e - 005$	$1.0731e - 002$	$2.2730e + 000$	$1.2854e + 000$	$2.5114e + 002$
Glass	Average	$2.2368e + 002$	$2.2539e + 002$	$2.5881e + 002$	$2.5239e + 002$	$2.3557e + 002$
	Best	$2.1232e + 002$	$2.1087e + 002$	$2.2911e + 002$	$2.2044e + 002$	$2.1574e + 002$
	Worst	$2.4627e + 002$	$2.5320e + 002$	$2.9908e + 002$	$3.0959e + 002$	$2.5538e + 002$
	Std	$7.8323e + 000$	$1.2685e + 001$	$1.4607e + 001$	$2.0131e + 001$	$1.2471e + 001$

Ripley's glass ($N = 214$, $d = 9$, $K = 6$), for which data were sampled from six different types of glass: building windows float processed (70 objects), building windows nonfloat processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects), and headlamps (29 objects), each with nine features, which are refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium, and iron [29].

For every data set, each algorithm is applied 30 times individually with random initial solution. The parameters of all algorithms are set like Section 4. Table 2 summarizes the intracluster distances, as defined in (5.3), obtained from all algorithms for the data sets above. The average, best, and worst solution of fitness from the 30 simulations, and standard deviation are presented in Table 2. Figure 2 shows the search progresses of the average values found by four SI algorithms over 30 runs for six data sets. Figure 3 shows the original data distribution of MotorCycle and Iris data sets and the clustering result by CABC algorithm.

From the values in Table 2, we can conclude that the results obtained by CABC are clearly better than the other algorithms for all data sets; CPSO is a little better than PSO; the K-means is the worst for all data sets.

For MotorCycle data set, the optimum of the fitness function for all algorithms, except K-means, is $2.0606e + 003$. That means they all can found the global solution. However, the standard deviations for them are $3.2158e - 001$, $1.9118e + 001$, $4.6354e + 001$, and $1.3340e + 001$,

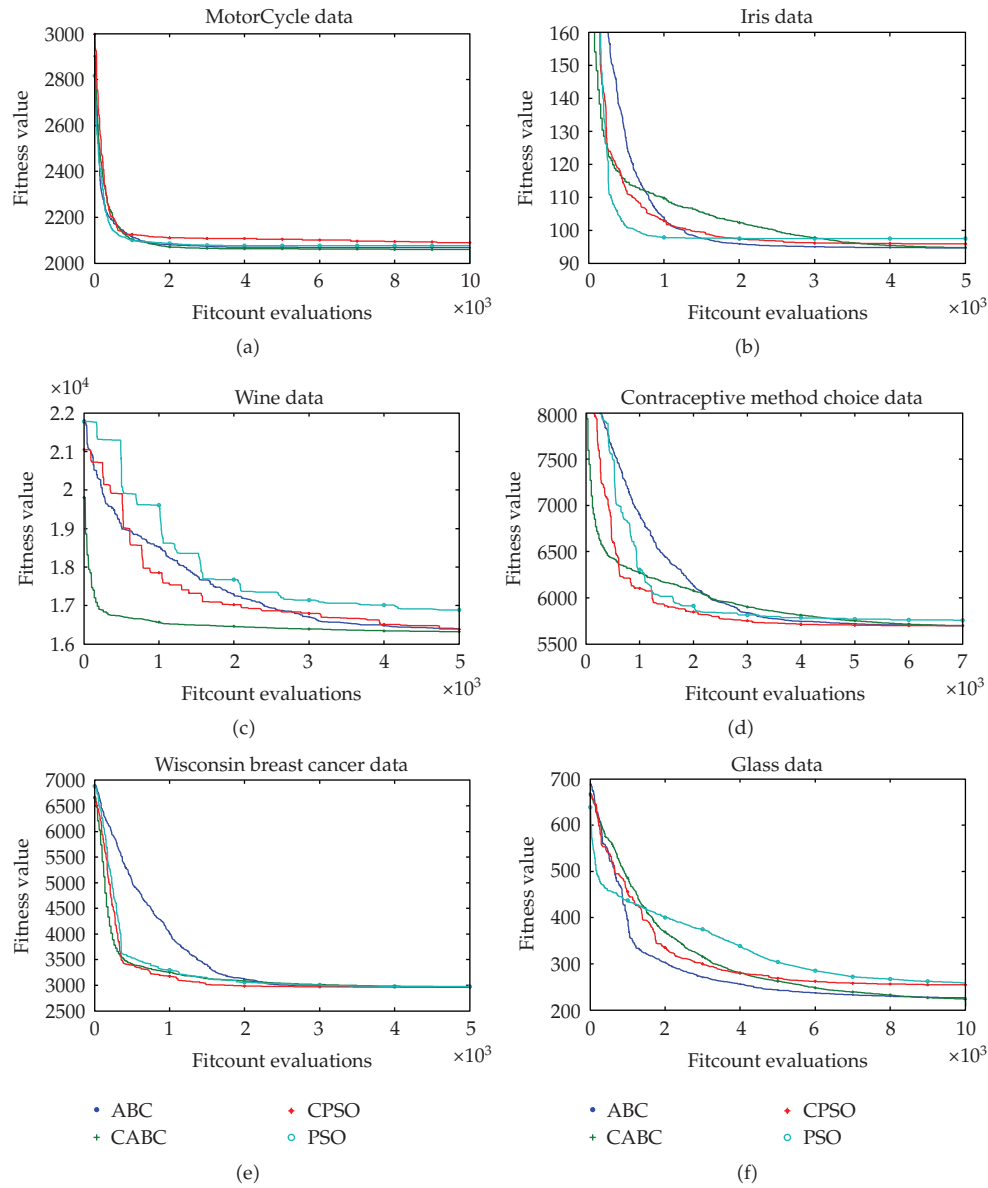


Figure 2: Comparing the convergence of the proposed CABC-based clustering with other approaches in terms of total mean-square quantization error (a) MotorCycle (b) Iris. (c) Wine. (d) CMC. (e) Cancer. (f) Glass.

respectively. From the standard deviation, we can see that the CABC algorithm is better than the other methods.

For Iris data set, CABC and ABC provide the optimum value and small standard deviation in compare to those of obtained by the other methods. The average values of the fitness function for CABC and ABC are $9.4603e +001$ and $9.4607e +001$, respectively; the standard deviations for CABC and ABC algorithms are less than 1. That means CABC and ABC converge to the global optimum most of the times. For the CABC, note that I adopted

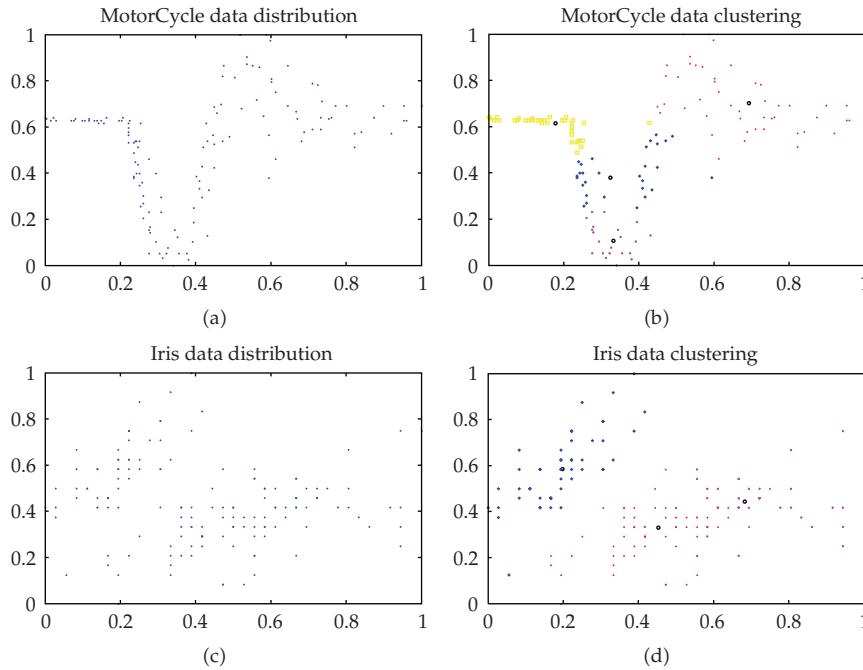


Figure 3: The original data distribution of MotorCycle and Iris data sets and the clustering result by CABC algorithm. (a) MotorCycle distribution (b) MotorCycle clustering result. (c) Iris distribution. (d) Iris clustering result.

scientific notation and I kept four digits after the decimal point. Actually, the best, worst and, average values are not exactly the same. They are just very close, so the standard deviation is very small, but it is different from zero.

For Wine data set, the optimum value, the average value, and the standard deviation of the fitness function for CABC and ABC are almost the same. The results of CABC and ABC algorithms are far superior to those of the other methods.

For CMC data set, the best global solution, the worst global solution, the average value and the standard deviation of the CABC are $5.6937e+003$, $5.6939e+003$, $5.6938e+003$, and $4.5501e-002$, respectively. The ABC algorithm is as good as CABC. Both of them significantly are smaller than the other methods.

For Wisconsin breast cancer data set, the averages of the fitness for CABC and ABC are almost identical to the best distance, and the standard deviation of the fitness for the CABC and ABC algorithms are $1.8380e-005$ and $1.0731e-002$, respectively. It means that the CABC and ABC algorithms are able to converge to the global optimum $2.9644e+003$ in all of runs, while *K*-means and PSO may be trapped at local optimum solutions.

Finally, for the Ripley's glass data set, Table 2 shows that the average, best, worst, and standard deviation values of the fitness function for CABC algorithm are much smaller than those of the other four methods. The CABC clustering algorithm is able to provide the same partition of the data points in all runs. From the result above, for all data sets, CABC outperforms the other four methods.

Clustering result of MotorCycle and Iris data sets by CABC algorithm are presented in Figure 3. The black ring is clustering center.

7. Conclusion

In this paper, based on the cooperative approaches, a novel Article Bee Colony (ABC) algorithm is presented, namely, Cooperative Article Bee Colony (CABC). Because of cooperation, the final solution is produced using information from all the populations. This results in a significant improvement in the performance in terms of solution quality and convergence speed. In order to demonstrate the performance of the CABC algorithm, we compared the performance of the CABC with those of ABC, PSO, and CPSO optimization algorithms on several benchmark functions. From the simulation results, it is concluded that the proposed algorithm has the ability to attain the global optimum; moreover, the CABC definitely outperforms the original ABC.

Because the CABC algorithm can be efficiently used for multivariable, multimodal function optimization, we apply it to solve clustering problems. The algorithm has been tested on several well-known real data sets. To evaluate the performance of the CABC algorithm on clustering problems, we compare it with the original ABC, PSO, CPSO, and k -means. From the experimental results, we can see that the proposed optimization algorithm is better than the other algorithms in terms of average value and standard deviations of fitness function. Additionally, the simulation result illustrates that the proposed CABC optimization algorithm can be considered as a viable and efficient method to solve optimization problems.

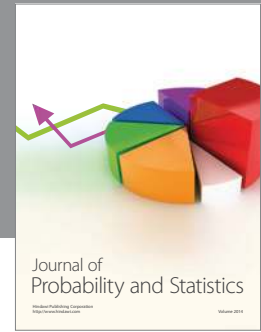
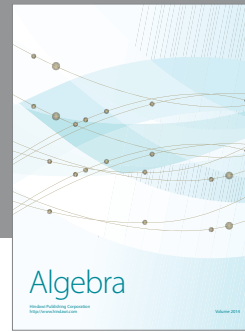
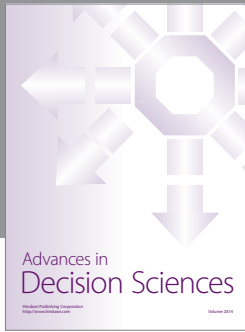
Acknowledgment

This work is supported by the 863 Hi-Tech research and development program of China under contract No. 2008AA04A105, and Natural Science Foundation of Liaoning Province under contract No. 20091094

References

- [1] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.
- [2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.
- [3] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.
- [4] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.
- [5] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [6] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Proceedings of the 12th International Fuzzy Systems Association World Congress on Foundations of Fuzzy Logic and Soft Computing (IFSA '07)*, vol. 4529 of *Lecture Notes in Artificial Intelligence*, pp. 789–798, Springer, 2007.
- [7] D. Karaboga and B. Akay, "Artificial bee colony (ABC) Algorithm on training artificial neural networks," in *Proceedings of the 15th IEEE Signal Processing and Communications Applications (SIU '07)*, pp. 1–4, June 2007.
- [8] C. Ozturk and D. Karaboga, "Classification by neural networks and clustering with artificial bee colony (ABC) algorithm," in *Proceedings of the 6th International Symposium on Intelligent and Manufacturing Systems, Features, Strategies and Innovation*, Sakarya, Turkey, October 2008.
- [9] M. A. Potter and K. A. de Jong, "A cooperative coevolutionary approach to function optimization," in *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*, pp. 249–257, Springer, Berlin, Germany, 1994.

- [10] F. van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.
- [11] H. Chen, Y. Zhu, and K. Hu, "Cooperative bacterial foraging optimization," *Discrete Dynamics in Nature and Society*, vol. 2009, Article ID 815247, 17 pages, 2009.
- [12] I. E. Evangelou, D. G. Hadjimitsis, A. A. Lazakidou, and C. Clayton, "Data mining and knowledge discovery in complex image data using artificial neural networks," in *Proceedings of the Workshop on Complex Reasoning on Geographical Data*, Paphos, Cyprus, 2001.
- [13] M. S. Kamel and S. Z. Selim, "New algorithms for solving the fuzzy clustering problem," *Pattern Recognition*, vol. 27, no. 3, pp. 421–428, 1994.
- [14] M. Omran, A. Salman, and A. P. Engelbrecht, "Image classification using particle swarm optimization," in *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning*, Singapore, 2002.
- [15] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Academic Press, New York, NY, USA, 2001.
- [16] B. Zhang and M. Hsu, "K-harmonic means—a data clustering algorithm," Tech. Rep. HPL-1999-124, Hewlett-Packard Labs, 2000.
- [17] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, NY, USA, 1981.
- [18] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An ant colony approach for clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187–195, 2004.
- [19] Y. Kao and K. Cheng, "An ACO-based clustering algorithm," in *Proceedings of the 5th International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS '06)*, M. Dorigo, Ed., vol. 4150 of *Lecture Notes in Computer Science*, pp. 340–347, Springer, Berlin, Germany, 2006.
- [20] M. Omran, A. P. Engelbrecht, and A. Salman, "Particle swarm optimization method for image clustering," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 3, pp. 297–321, 2005.
- [21] V. D. Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proceedings of IEEE Congress on Evolutionary Computation (CEC '03)*, pp. 215–220, Canberra, Australia, 2003.
- [22] D. Karaboga and C. Ozturk, "A novel clustering approach: artificial bee colony (ABC) algorithm," *Applied Soft Computing*, vol. 11, pp. 652–657, 2010.
- [23] C. L. Blake and C. J. Merz, "UCI Repository of Machine Learning Databases," <http://archive.ics.uci.edu/ml/datasets.html>.
- [24] B. Akay and D. Karaboga, "Parameter tuning for the artificial bee colony algorithm," in *Proceeding of the 1st International Conference on Computational Collective Intelligence (ICCCI '09)*, pp. 608–619, Wroclaw, Poland, 2009.
- [25] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007.
- [26] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '99)*, vol. 3, pp. 1945–1950, Piscataway, NJ, USA, 1999.
- [27] K. R. Žalik, "An efficient k-means clustering algorithm," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1385–1391, 2008.
- [28] Z. Güngör and A. Ünler, "K-harmonic means data clustering with simulated annealing heuristic," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 199–209, 2007.
- [29] T. Niknam, B. Bahmani Firouzi, and M. Nayeripour, "An efficient hybrid evolutionary algorithm for cluster analysis," *World Applied Sciences Journal*, vol. 4, no. 2, pp. 300–307, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

