International Journal of Control and Automation
Vol. 9, No. 4 (2016), pp. 107-116
http://dx.doi.org/10.14257/ijca.2016.9.4.11

An Optimized Artificial Bee Colony Algorithm for Clustering

An Gong*, Yun Gao, Xingmin Ma, Wenjuan Gong, Huayu Li and Zhen Gao

School of Computer & Communication Engineering, China University of Petroleum, Qingdao 266580, China
**gongan0328@sohu.com*

Abstract

K-means algorithm is sensitive to initial cluster centers and its solutions are apt to be trapped in local optimums. In order to solve these problems, we propose an optimized artificial bee colony algorithm for clustering. The proposed method first obtains optimized sources by improving the selection of the initial clustering centers; then, uses a novel dynamic local optimization strategy utilizing roulette wheel selection algorithm for further enhancing local optimization. To prove its effectiveness, we validate the proposed algorithm on four datasets from UCI and compared the results with K-means, K-means++ and Artificial Bee Colony algorithm. Experiment results show that the proposed algorithm performs better than other clustering algorithms.

Keywords:*Clustering, Artificial Bee Colony, K-means, Roulette wheel selection algorithm, Dynamic local optimization*

1. Introduction

Cluster algorithm plays a crucial part in learning inherent structure of dataset and is widely applied in image segmentation, data compression, data mining and so on. Its application domain ranges from psychology and biology to geology [1-3]. Among all, K-means [4] is one of the most classical and practical clustering algorithms, and has been widely applied in many areas thanks to its simplicity, efficiency, and local search ability. But k-means has the drawback of excessive dependence on the initial cluster centers, and is easy to fall in local optimums.

Lots of research has been done to solve the above-mentioned problems. For example, authors in [5] propose to measure distances between all existing cluster centers, and those which are further from selected cluster centers are more probable to be selected as new cluster centers. This guarantees that clustering centers are separated as far as possible, thus improving the quality of the initial cluster centers. Another example is in [6], which enhances clustering accuracies and stabilities by applying differential analytical method to k-dist graph and choose initial cluster centers as those with comparatively higher density. Authors in [7] propose an iterative method for finding initial cluster centers: first the largest cluster is found, then existing clusters are split into smaller clusters using two data points furthest from each other within a cluster, and the splitting procedure is repeated until a specified number of clusters are reached. All these methods improve clustering results by selecting the optimized initial cluster centers, but still their optimization might be trapped into local optimums.

Recently, swarm intelligence algorithm has been successfully applied in clustering problems due to its capabilities of searching for optimums globally [8-11]. Works in [12-13] employ particle swarm optimization to enhance K-means algorithm, and is applied in network intrusion detection system, significantly reducing the false alarm rate and improving the system performance. Authors in [14] propose a K-means method based on an

* Corresponding Author

ISSN: 2005-4297 IJCA

artificial fish swarm algorithm. It calculates distances between objects based on weighted attributes in information gain, according to which input data are divided into clusters and better clustering results are achieved. Artificial Bee Colony (ABC) algorithm [15] was originally proposed to solve function optimization problems. It has several advantages, for example there are only a few parameters, the principle is simple and it is easy to implement. What's more, it performs better in global regulations and local optimizations than genetic algorithms, differential evolution algorithm and particle swarm optimization algorithm [16-18]. But ABC algorithm has a slow convergence rate, and is not good at local search. In this paper, we propose to select the optimum initial sources based on which a dynamic adjustment to local search ranges is employed. The proposed algorithm alleviates the problems of the original ABC algorithm and achieves better clustering accuracies and stabilities.

2. Foundations

2.1. K-means Algorithm

K-means algorithm, a classic clustering algorithm based on division, is proposed by James MacQueen in 1967. The idea is to identify K cluster centers and a dataset is divided into K clusters accordingly, so that squared errors within a cluster (that is, the sum of squared distances between every sample point and the cluster center to which the sample point belongs) is minimum. Formally it can be denoted as:

$$U = \bigcup_{j=1}^{k} C_j, \tag{1}$$

$$C_j \neq \phi, \text{ and } \quad C_i \bigcap C_j (i \neq j) = \phi, \tag{2}$$

$$E = \sum_{j=1}^{k} \sum_{x_t \in C_j} d^2(x_t, C_j), \tag{3}$$

where $U = \{C_1, C_2, \ldots, C_k\}$ denotes a set of samples to be clustered, $C_j (i, j = \{1, 2, \ldots, K\})$ represents the last cluster after division, x_t denotes a sample point belonging to cluster C_j, and $d(x_t, C_j)$ denotes the distance between the sample point x_t and its cluster center C_j. E is a criterion function for measuring clustering results, calculated as the sum of squared errors within clusters. Generally, smaller E values mean better clustering results.

The pros of K-means algorithm includes: (1) high simplicity, fast convergence rate, excellence especially dealing with large datasets; (2) better performances for spherical clusters and dataset whose data samples from variant clusters show obvious differences. The cons of K-means algorithm are: (1) predetermined clustering number K; (2) performances' over reliance on the selection of initial cluster centers, tendencies of being trapped in local optimums, and poor robustness; (3) high sensitivity to noise and isolated sample points; (4) low effectiveness of processing non-spherical clusters.

2.2. Artificial Bee Colony Algorithm

The ABC algorithm is inspired by foraging behaviors of honey bee swarms and complies with two principles of a swarm intelligence behavior model: self-organization and division of labor. This behavioral model is consisted of four components: food sources, employed foragers (or employed bees), onlooker bees, and scout bees, of which onlookers and scout bees compose unemployed bees.

The position of a food source denotes a possible solution to the clustering problem. Scout bees carry out a global search for food sources by performing random searches and employed and onlooker bees search for neighbor solutions. Employed bees evaluate nectar qualities of food sources from memory. Onlooker bees on the dance area receive nectar information of food sources from employed bees and choose the best food source. A food

source is assumed to be abandoned if the position cannot be improved with a predefined number of cycles. And the abandoned source is replaced with a new source by scout bees. The outline of the algorithm is shown below:

Step 1. Initialization Step. Randomly generate *SN* initial solutions x_i ($i = \{1, 2, \ldots, SN\}$), each of which denotes a food source. And initialize *SN* employed bees and *SN* onlooker bees. Note that the random initialization formula is defined as below:

$$x_{ij} = x_{\min\ j} + rand\ (0,1)(x_{\max\ j} - x_{\min\ j}), \tag{4}$$

where $x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ is a vector of length D and x_{minj} and x_{maxj} denote the minimum and maximum entry value from the j^{th} dimension, respectively.

Step 2. Each employed bee finds new solutions according to the following formula:

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}), \tag{5}$$

where $v_i = \{v_{i1}, v_{i2}, \ldots, v_{iD}\}$ is a new solution within a local range of old sources, $x_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$ is an old source remembered from previous iteration, v_{ij} and x_{ij} ($i = \{1, 2, \ldots, SN\}$, $j = \{1, 2, \ldots, D\}$) denote entry values from the j-th dimension of v_i and x_i, respectively, φ_{ij} ($\varphi_{ij} \in (-1, 1)$) is a random real number and k ($k = \{1, 2, \ldots, SN\}$) is a randomly selected integer not equal to i. Then fitness values of all candidate sources are computed inversely proportional to the sum of Euclidean distances between sample points and their cluster centers. Finally, fitness values are compared between new and old sources and sources are selected with a greedy algorithm.

Step 3. Calculate probabilities p_i of the solution x_i as below:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \tag{6}$$

where fit_i is the fitness value of a solution x_i. According to the probabilities, each onlooker bee selects a new solution x_i. Then onlooker bees search for local optimums following Equation (5), calculate fitness value and apply greedy selection algorithms.

Step 4. After predefined number of iterations (denoted by limit), if employed and onlooker bees are not able to find a new source with better nectar through local search, this solution is abandoned and replaced with a new solution by scout bees. Scout bees search for new solutions with a random global selection.

Step 5. Repeat from Step 2, until maximum number of iterations (denoted by *MCL*) is reached, and output the solution.

In total, there are three parameters to be predefined in an ABC algorithm: food source number *SN*, abandonment threshold limit, and maximum iteration number *MCL*. And an ABC algorithm contains four selection procedures [17]: local election performed by employed and onlooker bees utilizing remembered information, searching within a local area, and selecting local optimums; greedy selection by employed and onlooker bees for comparing qualities of new and old sources, and select accordingly; global selection carried out by onlooker bees based on finesses; global, random selection by scout bees aimed at selecting new source globally. These four selection processes ensure that the algorithm has a better ability to control globally and locally search; it reaches comparatively better balance; and greatly enhances the performance of the algorithm.

3. An Optimized Artificial Bee Colony Algorithm for Clustering

ABC algorithm randomly generates initial food sources and local search ranges, resulting in uneven distribution of initial solutions, poor local convergence, and tendencies to fall in local optimums. We propose a new food source initialization method and a dynamic local search strategy based on K-means++ and roulette wheel selection algorithm. The

proposed methods are utilized alternately with K-means algorithm to obtain optimal divisions in clustering.

3.1. Optimize The Initial Food Sources

The ABC algorithm utilizes Equation (4) to initialize food sources with random selections, but since clustering problems have its own characteristics, different initializations result in great differences in clustering results, one of the drawbacks of the clustering algorithms based on division. Inspired by K-means++ algorithm[5], we choose initial sources with a selection method based on distance and roulette wheel selection algorithm, so as to ensure that cluster centers of food sources are as far as possible from each other. Suppose the dataset is to be divided into K clusters, the proposed single source initialization algorithm is described as follows:

Step 1. Choose one center uniformly at random from among the data points.

Step 2. For each data point x, compute $D(x)$, the distance between x and the nearest center that has already been chosen.

Step 3. Choose new cluster centers based on distance measure $D(x)$. The selection is based on the following rules: a sample point x is more probable to be chosen as a new cluster center if it is farther to existing cluster centers (i.e., with a larger $D(x)$).

Step 4. Repeat from Step 2 until a total cluster center number K is reached, and output the cluster centers as food sources.

Note that, to make sure that the newly chosen cluster center in Step 3 is not an isolated data point, we don't choose the furthest sample point directly as the new cluster center but use a roulette wheel selection algorithm for selection.

3.2. Dynamic Local Optimization Strategies

The original ABC algorithm employs Equation (5) for employed and onlooker bees searching for local optimal food sources. This methodology adopts a random selection strategy in a local area, but doesn't consider the need for using a varying search range in each iteration, resulting in low convergence rate. Authors in [18] point out that in the early stage of ABC algorithm, employed and onlooker bees should explore larger areas, so as to quickly localize better food sources; as the search procedure progresses, bee swarms should be able to adjust the search range, and with a finer search, the converge rate could be speeded up and a better convergence is able to be achieved.

To achieve this goal, we propose a dynamic search strategy derived from roulette wheel selection algorithm. The proposed algorithm takes optimized food sources from previous step and considers iteration numbers to improve the situation of slow convergence rate and poor local search ability. From Equation (5) we see that local search range is determined from φ_{ij}, x_{kj} and x_{ij}: the bigger φ_{ij} and the bigger difference between x_{kj} and x_{ij}, the larger is the search range; and vice versa. Thus, we optimize from these parameters.

In the original ABC algorithm, the size of x_{kj} depends on the predetermined value of K, and K is generated randomly, as a result the algorithm is not able to control search ranges effectively, and this weaken its local search ability[18]. After initializing food sources with the method described in section 3.1, cluster centers of food sources are chosen to be away from each other as far as possible, and we further utilize distances between cluster centers as local search range. Suppose x_{imax} and x_{imin} are the furthest and nearest sample points from cluster center S_i in cluster S, let M denote the distance between x_{imax} and x_{ij}, and N denote the distance between x_{imin} and x_{ij}, we choose M as a large search range and (M-N) as a small search range. By this definition, in every iteration of the ABC clustering algorithm, each cluster center is approaching its optimal division, that is, the local search range is varied in each iteration, thus ensuring its abilities of dynamic search for optimums.

In the original ABC algorithm, φ_{ij} is generated randomly; as a result the algorithm is not able to adjust search range of a bee swarm in its search for optimums accordingly, decreas-

ing convergence rate. In our method, based on roulette wheel selection algorithm, we propose a strategy with dynamic search range: in early stage of the algorithm, a bee swarm searches in a wide range *M* with a higher probability; later on, a bee swarm adjusts its search range and search in a smaller range (*M-N*) with a higher probability. Suppose iteration number is denoted by *g* (*g* = {*1*, *2* , …, *MCL*}), and *MCL* denotes maximum iteration number, we outline the dynamic search algorithm as follows:

Step 1. Set *p* as *0* or *1*. *P0* (that is, when *p* is set as *0*) denotes search in a small range (*M-N*), and *P1* (that is, when *p* is set as *1*) denotes search in a wide range *M*.

Step 2. Suppose the fitness of *P0* is *g*, and the fitness of *P1* is *MCL-g*, the overall fitness is *MCL*.

Step 3. The probability of *P0* is $p(P0) = \frac{g}{MCL}$, and the probability of *P1* is $p(P1) = \frac{MCL - g}{MCL}$.

Step 4. Construct a roulette based on the cumulative probabilities of *P0* and *P1*.

Step 5. Apply roulette selection algorithm: randomly select a real number in the range of [*0*, *1*], choose *P0* if the number falls in the range of its cumulative probabilities and *P1* otherwise.

Based on the described method, we improve the local search strategy of ABC algorithm and its local search formula is described as below:

$$v_{ij} = x_{ij} + (pM + (1-p)(M-N)) \times rand(-1,1) \quad (7)$$

where v_i and x_i denote new and old sources, respectively. *P* is determined by the search strategy and set as *0* or *1* accordingly.

We observe from the above steps that: in the early stage of the algorithm, *g* is relatively small, so *p* is highly probable to be selected as *1*, resulting bees searching in a wide range *M* with high probabilities; while in later stages of the algorithm, *g* increases gradually and *p* is more probable to be selected as *0*, thus bees are apt to search within a smaller range of (*M-N*). Therefore, the proposed method is able to meet the demand of local search adjustment in optimization process and improve the efficiency and quality of convergence.

3.3. Algorithm Description

We describe the optimized artificial bee colony algorithm for clustering as follows:

Step 1. Source initialization. Initialize food sources with the method described in ***section 3.1***.

Step 2. Each employed bee employs the dynamic local search strategy proposed in ***section 3.2***, and selects food sources with greedy selection algorithm.

Step 3. Each onlooker bee chooses food sources through roulette wheel selection algorithm and then adopts a local search strategy proposed in ***section 3.2*** and updates food sources by applying greedy selection algorithm. If a food source couldn't be improved through *limit* number of iterations, it is abandoned and its bee is transformed into a scout bee.

Step 4. Each scout bee adopts an optimization strategy described in ***section3.1*** for selecting new food sources.

Step 5. Repeat from ***Step 2*** until iteration number reaches *MCL*, and output the clustering results.

4. Experimental Results

We evaluate the proposed method on Iris, Wine, Glass and Seeds datasets from UCI [19], as shown in 0.

Table 1. Comparisons of Datasets for Experiments

Dataset	Number of attributes	Number of clusters	Data set size
Iris	4	3	150（50,50,50）
Wine	13	3	178（59,71,48）
Glass	10	7	214（70,76,17,0,13,9,29）
Seeds	7	3	210（70,70,70）

For comparison, error rate is adopted as the clustering criterion. Error rate refers to the number of misclassified samples over the total number of samples, formulated as:

$$Err = \frac{\varepsilon}{n} \times 100\ \%, \tag{8}$$

We compare the proposed method with K-means algorithm, K-means++ algorithm and the original ABC algorithm. As the algorithms use random selections in certain part resulting in uncertainty in experiments, we run experiments 20 times for each algorithm and take the average as the final result. The initial parameters for ABC algorithm are set following: food source number *SN* equals *20*, abandonment threshold *limit* equals *SN*D*, the maximum number of iterations *MCL* equals 20, and use *D* to represent the number of attributes in each dataset. We further split the experiments of the proposed methodology into two steps: first is experiments of the algorithm based on optimized food sources (named IABC-1); second is experiments of algorithm further adding dynamic local search strategies (named IABC-2). Experimental results are shown in 0, Avg_{Err}, Max_{Err} and Min_{Err} refer to the average, maximal, minimum value of the error rate respectively.

As can be seen from the experimental results: K-means++ performs slightly better than K-means in general; ABC algorithm sometimes performs good and sometime performs bad; the algorithm proposed in this paper fully utilizes the advantages of all above algorithms, reduces clustering errors effectively resulted from random selections, and improves accuracies and stability to some extent.

We compare the running time of ABC based algorithm, as shown in 0We can see that the proposed algorithm outperforms the original ABC algorithm in efficiency. Algorithm IABC-1 is relatively more efficient than algorithm IABC-2 while algorithm IABC-2 provides more accurate results. Due to its simplicity, K-means algorithm performs faster than all ABC based method but is much less accurate.

To give a straightforward impression of the clustering results, we visualize clustering results of the K-means algorithm and the proposed method on Iris dataset, shown in Figure 2. As can be seen, the proposed algorithm is more reasonable in dividing this dataset, and achieves good clustering results.

Table 2. Accuracy Comparison of Clustering Algorithms on Four Standard Datasets

Dataset	Algorithm	Avg_{Err}	Max_{Err}	Min_{Err}
Iris	K-means	12.43%	18.00%	10.67%
	K-means++	11.13%	11.33%	10.67%
	ABC	10.70%	11.33%	9.33%
	IABC-1	**10.50%**	11.33%	9.33%
	IABC-2	**8.71%**	9.33%	7.33%
	K-means	28.60%	30.34%	25.84%

Wine	K-means++	27.81%	29.78%	25.84%
	ABC	32.30%	33.15%	32.02%
	IABC-1	**23.60%**	26.97%	20.22%
	IABC-2	**15.73%**	19.66%	12.36%
Glass	K-means	39.25%	44.86%	26.64%
	K-means++	35.87%	35.98%	35.05%
	ABC	44.63%	45.79%	43.93%
	IABC-1	**28.04%**	33.18%	25.70%
	IABC-2	**23.65%**	28.97%	15.89%
Seeds	K-means	10.71%	10.95%	10.48%
	K-means++	10.60%	10.95%	10.48%
	ABC	10.35%	11.43%	9.52%
	IABC-1	**10.00%**	10.47%	7.14%
	IABC-2	**8.81%**	10.00%	6.19%

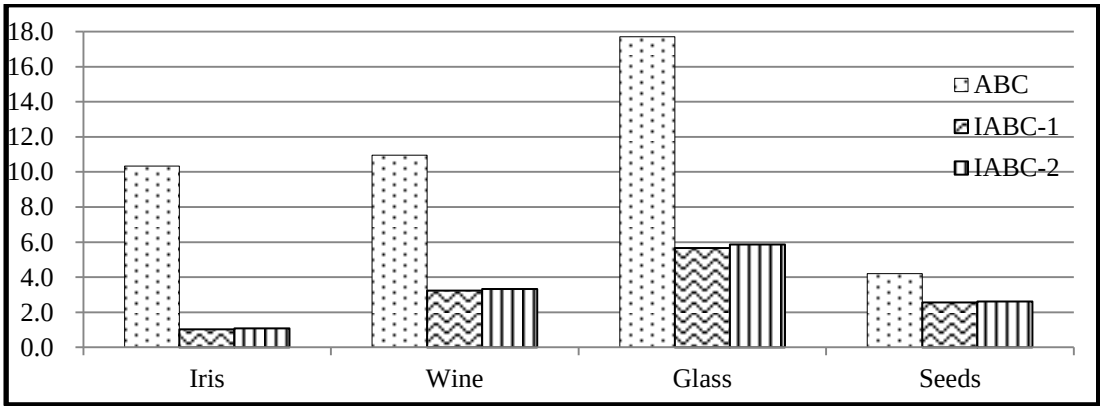
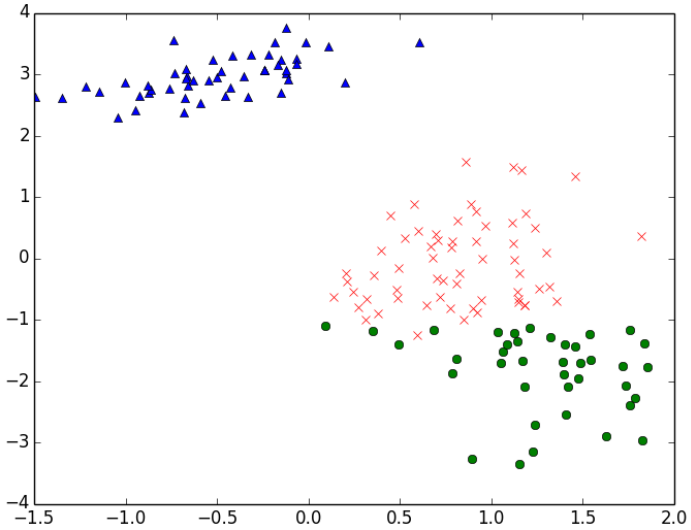


Figure 1. Efficiency Comparison of ABC Based Algorithms



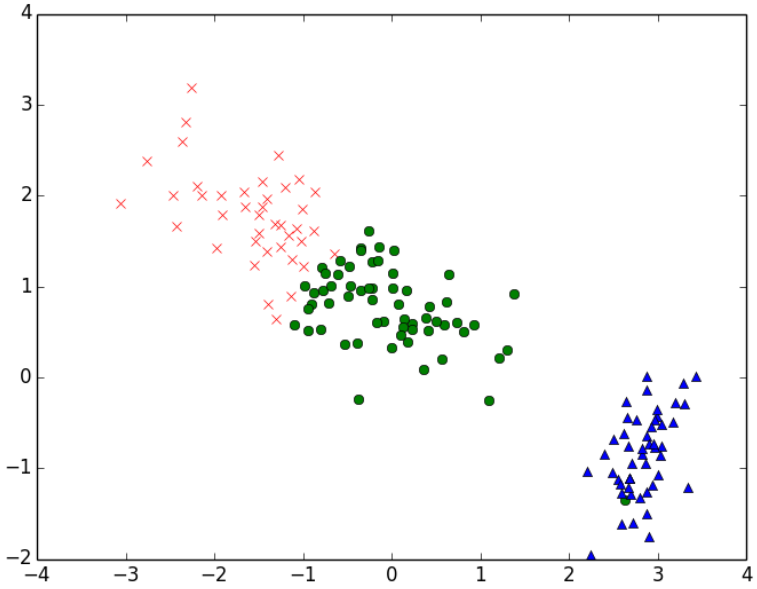


Figure 2. Clustering Results Of K-Means(Above) and the Optimized Artificial Bee Colony Algorithm(Below) on Iris Dataset

5. Conclusions

In this paper, we studied optimized initialization and dynamic local search range strategy for ABC algorithm, proposed an optimized artificial bee colony algorithm for clustering, thus enhancing the overall performance of the algorithm. In the experiments with four datasets from UCI, the proposed algorithm is less sensitive to initial clustering centers, and has higher accuracies. In the future work, we would like to explore how to search more efficiently.

Acknowledgment

This work is supported by Natural Science Foundation of Shandong Province 2012 "Formation Mechanism and Control Technology Research on Methane Gas Hydrate Containing Acid Gases" (with number: ZR2012EEM020) and the Fundamental Research Funds for the Central Universities (with number: 14CX02030A).

References

[1] A. K. Jain and P. J. Flynn. "Image segmentation using clustering", Ahuja N, Bowyer K, eds. Advances in Image Understanding: A Festchrift for Azriel Rosenfeld. Piscataway: IEEE Press, **(1996)**. p p. 65-83.
[2] I. Cades, P. Smyth and H. Mannila, "Probabilistic modeling of transactional data with applications to profiling, visualization and prediction, sigmod", Proc. of the 7th ACM SIGKDD, San Francisco: ACM Press, http://www.sigkdd.org/kdd2001/, **(2001)**. pp. 37-46.
[3] A. K. Jain, M. N. Murty and P. J. "Flynn. Data clustering: A review", ACM Computing Surveys, vol. 31, no. 3, **(1999)**, pp. 264-323.
[4] A. K. Jain, "Data clustering: 50 years beyond K-means", Pattern Recognition Letters, vol. 31, no. 8, **(2010)**, pp. 651-666.
[5] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding", Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, **(2007)**, pp. 1027-1035.
[6] D. Zheng and Q. Wang, "Selection algorithm for the initial clustering center in K-means", Journal of Computer Applications, vol. 8, **(2012)**, pp. 2186-2188.
[7] G. Chen, W. Wang and J. Huang, "A K-means algorithm based on enhanced selection of initial clustering centers", Journal of Chinese Computer System, vol. 6, **(2012)**, pp. 1320-1323.
[8] X. Cui and T. E. Potok, "Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm", Journal of Computer Sciences (Special Issue), **(2005)**, pp. 27-33.

[9] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm", AppliedMathematics and Computation , vol. 214, **(2009)**, pp. 108-132.
[10] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", Applied Soft Computing, vol. 11, **(2011)**, pp. 652-657.
[11] K. K. Mizooji, A. T. Haghighat and R. Forsati, "Data clustering using bee colony optimization", 7th International Multi-Conference on Computing in the Global IT, **(2012)**, pp. 189-194.
[12] T. Fu and Y. Sun, "PSO based K-means algorithm and its application in detecting network invasion", Computer Science, vol. 5, **(2011)**, pp. 54-55.
[13] T. Fu and W. Sun, "PSO-based K-means algorithm and its application in detecting network invasion", Computer Science, vol. 11, **(2013)**, pp. 137-139.
[14] H. T. Yu, M. Jia, H. Wang and G. Shao, "Enhanced K-means clustering algorithm based on artificial fish-swarm model", Computer Science, vol. 12, **(2012)**, pp. 60-64.
[15] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, **(2005)**.
[16] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", Applied soft computing, vol. 8, no. 1, **(2008)**, pp. 687-697.
[17] 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, **(2007)**, pp. 459-471.
[18] B. Akay and D. Karaboga, "A modified artificial bee colony algorithm for real-parameter optimization", Information Sciences, vol. 192, **(2012)**, pp. 120-142.
[19] UCI.Date sets. http://archive.ics.uci.edu/ml/datasets.html.

