Decision Science Letters 7 (2018) 65–76

Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

A novel hybrid K-means and artificial bee colony algorithm approach for data clustering

Ajit Kumar[a*], Dharmender Kumar[b] and S.K. Jarial[c]

[a]*Department of Computer Science and Engineering, DeenbandhuChhotu Ram University of Science and Technology, Murthal, India*
[b]*Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar, India*
[c]*Department of Mechanical Engineering, DeenbandhuChhotu Ram University of Science and Technology, Murthal, India*

C H R O N I C L E

Article history:
Received September 16, 2016
Received in revised format: October22, 2016
Accepted April15, 2017
Available online April172017

Keywords:
Artificial bee colony
Data clustering
F-measure
K-means
Objective function value
Tournament selection

A B S T R A C T

Clustering is a popular data mining technique for grouping a set of objects into clusters so that objects in one cluster are very similar and objects in different clusters are quite distinct. K-means (KM) algorithm is an efficient data clustering method as it is simple in nature and has linear time complexity. However, it has possibilities of convergence to local minima in addition to dependence on initial cluster centers. Artificial Bee Colony (ABC) algorithm is a stochastic optimization method inspired by intelligent foraging behavior of honey bees. In order to make use of merits of both algorithms, a hybrid algorithm (MABCKM) based on modified ABC and KM algorithm is proposed in this paper. The solutions produced by modified ABC are treated as initial solutions for the KM algorithm. The performance of the proposed algorithm is compared with the ABC and KM algorithms on various data sets from the UCI repository. The experimental results prove the superiority of the MABCKM algorithm for data clustering applications.

1. Introduction

Clustering is an important unsupervised classification technique to discover hidden patterns or information from a given dataset. It is a process of partitioning a set of objects/data into disjointed groups called clusters such that objects similar to each other are identified in one cluster (different from those in other clusters). Similarity may be expressed in terms of Euclidean distance; the lesser the distance, the more similarity is there between two objects or two clusters. Clustering is widely used in numerous applications. K-means (KM) algorithm is the most widely used clustering algorithm due to its efficiency in clustering of large data sets and faster convergence. It generates partitions of an N-dimensional population such that each partition is having small within-class variance. In K-means, each cluster has a center called mean and attempt is made to minimize its objective function (a square error

* Corresponding author. Tel: +91-9416133810
E-mail address: ajit.hisar@gmail.com (A. Kumar)

doi: 10.5267/j.dsl.2017.4.003

function). However, it has some limitations such as dependence on initialization of cluster centers, sensitivity to outliers, non-guaranteed optimal solutions, and formation of unbalanced clusters.

In the past, several approaches based on natural phenomena, insects and swarms have been used to solve clustering problems. These methods include genetic algorithm (Murthy & Chowdhury, 1996), tabu search (Al-Sultan, 1995), simulated annealing (Selim&Alsultan, 1991), ant colony optimization (Shelokar et al., 2004), particle swarm optimization (Chen & Ye, 2004), artificial bee colony (Karaboga&Ozturk, 2011), cat swarm optimization (Santosa&Mirsa, 2009), teacher learning based optimization (Satapathy&Naik, 2011), charge system search algorithm (Kumar &Sahoo, 2014), and so on. Artificial Bee Colony is a population based algorithm introduced by Karaboga in 2005 and has been successfully used in a wide variety of applications (Karaboga, 2005).

In order to overcome the initial cluster centers dependency problem of KM, a hybrid algorithm (MABCKM) using the modified ABC and KM is proposed. The proposed algorithm incorporates the output of the modified ABC algorithm as initial solutions of KM algorithm. The experimental results on several data sets prove that the proposed algorithm is better than others in terms of efficiency and convergence speed. Section 2 covers the discussion on clustering, including KM algorithm. Section 3 briefly describes the ABC algorithm. Section 4 first describes the modifications in ABC algorithm, followed by the MABCKM algorithm for clustering problems. Section 5 illustrates the data sets and experimental results. Finally, Section 6 concludes the paper.

2. Clustering

Clustering is considered as an important component of data mining, a process to extract useful information by exploring and analyzing large amount of data through automatic or semiautomatic means. Clustering methods identify groups or clusters of a data set using a step by step approach in the sense that in each cluster there are objects similar to each other i.e. homogeneity within the clusters, yet different from those in other clusters i.e. heterogeneity between the clusters. Clustering may be performed in two ways: hierarchical clustering and partitional clustering. Both of these methods are treated as hard (or crisp) in nature as they work by assigning the data points to one cluster only. The partitional clustering algorithms divide the set of data objects into clusters by iteratively relocating objects without hierarchy (Gan et al., 2007). The clusters are gradually improved to ensure high quality of clustering. The center-based clustering algorithms are the most common partitional algorithms and have been extensively used in the literature.

2.1. Center-based clustering

The center-based clustering algorithms are very effective in handling large and high-dimensional databases. The most popular of these algorithms is K-means algorithm and is simple in nature. K-means is a form of hard partitional clustering as each data point is assigned to one cluster only. In KM, the process of assigning data objects to the disjoint clusters repeats until there is no significant change in objective function values or membership of clusters. The objective function for a set of data objects $X = \{x_1, \dots, x_N\}$, having K disjoint subsets C_j is given as:

$$KM(X,C) = \sum_{i=1}^{N} MIN\left\{\left\|x_i - c_j\right\|^2 \middle|\ j = 1, \dots, K\right\} \quad (1)$$

K-means is useful in terms of efficiency in clustering of large data sets as its complexity is proportional to the data set. Also, it tends to converge fast as it requires few function evaluations. However, the KM does not guarantee optimal solutions although it converges into good solutions.

3. Artificial Bee Colony Algorithm

ABC is a population based optimization algorithm which is iterative in nature. Basically, ABC consists of five phases: initialization phase, employed bees phase, probabilistic selection phase, onlooker bees phase and scout bees phase. Bees going to a food source already visited by them are employed bees while the bees looking for a food source are unemployed. The onlooker bees wait for the information

from employed bees for food sources and scout bees carry out search for new food sources. The information exchange among bees takes place through waggle dance. There is one employed bee for every food source. The main steps of ABC are as under:

3.1. Initialization phase

The locations of food sources are randomly initialized within the range of boundaries according to Eq. (2) given by:

$$x_{ij} = x_j^{min} + rand(0,1)\left(x_j^{max} - x_j^{min}\right), \tag{2}$$

where $i = 1, \ldots, SN \; and \; j = 1, \ldots, D$. SN indicates the number of food sources and taken as half of the bee colony, D is dimension of the problem, x_{ij} represents the parameter for i^{th} employed bee on j^{th} dimension, x_j^{max} and x_j^{min} are upper and lower bounds for x_{ij} .

3.2. Employed bee phase

Each employee bee is assigned to the food source for further exploitation. The resulting food source is generated according to Eq. (3) as given by:

$$v_{ij} = x_{ij} + \phi(x_{ij} - x_{kj}), \tag{3}$$

where k is a neighbor of i, $i \neq k$, ϕ is a random number in the range [-1,1] to control the production of neighbor solutions around x_{ij}, v_{ij} is the new solution for x_{ij}. The fitness of new food source is now calculated using Eq. (4) as below:

$$fit_i = \begin{cases} \dfrac{1}{1 + f_i} & , f_i \geq 0 \\ 1 + abs\,(f_i), & f_i < 0 \end{cases} \tag{4}$$

where f_i is the objection function associated with each food source and fit_i is the fitness value.

3.3. Probabilistic selection phase

For each food source a probability value is calculated using Eq. (5) as given below, and an onlooker bee selects the food source according to this value.

$$p_i = \frac{fit_i}{\sum_{j=1}^{N} fit_j}, \tag{5}$$

where p_i is selection probability of i^{th} solution.

3.4. Onlooker bee phase

Each onlooker bee selects a food source to exploit according to the probability associated with it (i.e. more fitness, higher the probability). The chosen food sources are exploited for better solutions using Eq. (3) and their fitness values are calculated using Eq. (4).

3.5. Scout bee phase

If a food source does not produce better solutions even up to a predefined limit, the food source is abandoned and the corresponding bee becomes a scout bee. A new food source is randomly generated in the search space using Eq. (2).

4. Proposed Modified ABCKM Algorithm

ABC algorithm is simple, robust in nature, good in exploration and easy to implement since it utilizes the adaptable features of honeybee swarm. However, it has shortcomings such as slow convergence, tendency to local optima traps in solving complex multimodal problems, poor exploitation to find food sources in solution search equation. Therefore, there is scope of enhancement in exploitation ability as well as convergence speed of the ABC algorithm by use of one or more techniques. A few modifications have been proposed in different phases of the original ABC algorithm in order to obtain better results.

4.1. Modifications in original ABC algorithm

The proposed modifications are:

1.Use of chaotic sequences in combination with opposition based learning in the initialization phase to generate better initial solutions.
2.Replacing the roulette wheel selection mechanism by variable tournament selection and replacing the worst solution by a random better solution, in the onlooker bee phase.

4.1.1. Initialization phase

In the standard ABC, the random initialization of population may affect the convergence characteristics and quality of solutions. In order to improve the convergence characteristics and population diversity, chaotic sequences have been successfully used instead of random sequences (Alatas, 2010). These sequences in combination with opposition based learning method generate better initial solutions (Rahnamayan et al., 2008). Based on these techniques, the algorithm for population initialization is given below (Gao& Liu, 2012):

1. Initialize the maximum number of chaotic iterations $K \geq 300$, the population size SN and the individual counter i=1, j=1
2. ***for*** $i = 1$ to SN do
3. ***for*** $j = 1$ to D do
4. Randomly initialize the chaotic variable $ch_{0,j} \in (0,1)$, iteration counter $k = 0$
5. ***for*** $k = 1$ to K do
6. Generate the updated chaotic variable according to the selected map i.e. sinusoidal iterator i.e. $ch_{k+1,j} = \sin(\pi ch_{k,j})$
7. ***end for***
8. Calculate the values of Z

$Z_{i,j} = x_{min,j} + ch_{k,j}(x_{max,j} - x_{min,j})$

9. ***end for***
10. ***end for***
11. Set the counter i=1, j =1
12. ***for*** i = 1 to SN do
13. ***for*** j = 1 to D do
14. Calculate the value of OZ using opposition based learning $OZ_{i,j} = x_{min,j} + x_{max,j} - Z_{i,j}$, $where \quad i = 1,2,\dots,SN \quad j = 1,2,\dots,D$
15. ***end for***
16. ***end for***
17. Select SN fittest individuals as initial population from the set $\{Z(SN) \cup OZ(SN)\}$.

4.1.2. Onlooker bees phase

Two modifications have been proposed in this phase to improve the quality of solutions. First step is to replace the roulette wheel selection mechanism by varying tournament selection mechanism. The size

of tournament is selected on the basis of population size and cycle number. The tournament selection scheme works by holding a tournament of TN individuals chosen from the population, where TN is taken as tournament size (Blickle& Thiele, 1995; Miller & Goldberg, 1995). A tournament size TN=2 is chosen in early stages for better exploration and a variable tournament size based on the current cycle number is chosen in later stages for better exploitation as given below. In second step, the worst solution is replaced by a randomly generated better solution in order to enhance quality as well as convergence speed.

If $SN \geq 20$, the tournament size is taken as:

$$TN = SN * \frac{i}{10}, \text{ if } MCN \times \frac{i-1}{10} \leq cycle \leq MCN \times \frac{i}{10} \text{ and } i = 1,...,10 \tag{6}$$

If $10 < SN < 20$, then tournament size is taken as:

$$TN = \begin{cases} 2 & ,\text{if } cycle \leq \dfrac{MCN}{5} \\ TN + \dfrac{SN - \text{mod}(SN,5)}{5} & ,\text{if } \dfrac{MCN}{5} < cycle \leq \dfrac{MCN}{5} \times 4 \text{ and } TN < SN \\ SN & ,\text{if } \dfrac{MCN}{5} \times 4 < cycle \leq MCN \end{cases} \tag{7}$$

If $SN \leq 10$, then tournament size is taken as:

$$TN = \begin{cases} 2 & ,\text{if } cycle \leq \dfrac{MCN}{5} \\ TN + 1 & ,\text{if } \dfrac{MCN}{5} < cycle \leq \dfrac{MCN}{5} \times 4 \text{ and } TN < SN \\ SN & ,\text{if } \dfrac{MCN}{5} \times 4 < cycle \leq MCN \end{cases} \tag{8}$$

where SN : number of employed or onlooker bees, TN: tournament size and MCN: maximum cycle number.

For small population, the tournament size is incremented by 1 to find solutions. However, with the growth in population this increment will slow down the algorithm and hence, the tournament size becomes dependent on current cycle. The high fitness food sources within this tournament size are chosen by the onlooker bees thus speeding up the algorithm. Moreover, the replacement of worst fitness solution by a randomly generated solution provides the scope for better quality of solutions.

4.2.Proposed MABCKM algorithm

We propose a hybrid algorithm, called MABCKM, based on modified ABC and KM for clustering problems. The hybrid algorithm incorporates the merits of KM and MABC to enhance the fitness value of each particle. Each particle is taken as a real numbered vector of dimensions K×D, K is the number of clusters and D is the dimension of data set. The fitness of each particle is evaluated using Eq. (1) i.e. smaller is the objective function value, higher is the fitness. The pseudo code for MABCKM is given below:

1.(Initialization phase)

Initialize the parameters including number of food sources SN, limit, maximum cycle number MCN, and current cycle number CN=0;

Initialize the food sources using modified initialization phase given in Section 4.1.1;

Evaluate the fitness of food sources using Eq. (1);
Send the employed bees to the current food source;
2.While (CN<=MCN) do
3.(Employed bee phase)
for (each employed bee)
Find a new food source in the neighborhood of old food source using Eq. (3);
Evaluate the fitness of new food source using Eq. (1);
Apply greedy selection on the original food source and the new one;
end for
4. (Probabilistic selection phase)
Calculate the probability values P_i for each food source using Eq. (5);
5.(Onlooker bee phase)
t=1;
while (current onlooker bee t<= SN)
Calculate the tournament size based on population using Eq. (6) or (7) or (8);
Out of the chosen tournament, find the food source having maximum probability value;
Generate new solution for the selected food source using Eq. (3);
Evaluate the fitness of new food source using Eq. (1);
Apply greedy selection on the original food source and the new one;
t=t+1;
end while
Replace the worst fitness food source with a randomly produced food source using Eq. (2); generate new solution using Eq. (3) and evaluate fitness value using Eq. (1), apply greedy selection on the original food source and the new one;
6.(Scout bee phase)
If (food source is not upgraded up to the limit)
Send a scout bee to the solution of food source produced using Eq. (2);
end if
7.Memorize the best solution obtained so far
CN=CN+1;
8.end while
9.The best solution obtained is taken as initial solution for the K-means;
10. Apply the K-means algorithm to obtain and evaluate the better solutions until the termination criteria is satisfied;
11. Output the final cluster centers

5. Experimental Results and Analysis

Six data sets are employed to test our proposed algorithm. The six data sets taken from UCI Machine Repository are iris, glass, lung cancer, soyabean (small), wine and vowel data sets. The summary of clusters, features and data objects in each data set are given in Table 1. We evaluate and compare the performance of KM, ABC and MABCKM algorithms in terms of objective function of KM algorithm. The quality of the respective clustering is compared using the following six criteria:

- The objective function value (OFV) of the KM algorithm i.e. $KM(X,C)$. Clearly, the smaller the value of objective function is, the higher the quality of clustering.
- The F-measure using the ideas of precision and recall from information retrieval (Dalli, 2003; Handl et al., 2003). For each class i and cluster j, precision and recall are then defined as $p(i,j)$ and $r(i,j)$ in Eq. (9), and the corresponding value under the F-measure is as Eq. (10), where we choose $b=1$ to obtain equal weighting for precision and recall. The overall F-measure for the data set of size N is given by Eq. (11). Obviously, the bigger F-measure is, the higher the quality of clustering is.

$$p(i,j)=\frac{n_{ij}}{n_j},\quad r(i,j)=\frac{n_{ij}}{n_i} \tag{9}$$

$$F(i,j) = \frac{(b^2+1).p(i,j).r(i,j)}{b^2.p(i,j)+r(i,j)} \tag{10}$$

$$F = \sum_{i=1}^{k} \frac{n_i}{N}.\max_j\{F(i,j)\} \tag{11}$$

- Silhouette index (Sil) to measure how well each data point lies within its cluster (Xu et al., 2012). The average distance of a data point x_i belonging to a cluster c_j from all other data points in c_j is taken as a_i. For other clusters c_h, with $h = 1, \ldots, k$ and $h \neq i$, the smallest average distance of x_i to all data points in c_h is taken as b_i in Eq. (12). The silhouette value of x_i is now calculated as s_i in Eq. (13). The overall Silhouette index is defined as average of s_i over all data points, as given in Eq. (14). Clearly, a larger Sil value indicates good quality of clustering.

$$a_i = (1/N_j - 1)\sum_{\substack{l=1,\ldots,N_j \\ l\neq i}} \|x_i - x_l\|, \qquad b_i = \min_{\substack{h=1,\ldots,k \\ h\neq j}} (1/N_h) \sum_{x_l \in c_h} \|x_i - x_l\| \tag{12}$$

$$s_i = \frac{b_i - a_i}{\max(a_i,\ b_i)} \tag{13}$$

$$Sil = \frac{1}{N}\sum_{i=1}^{N} s_i \tag{14}$$

- Adjusted Rand Index (ARI) assumes the generalized hypergeometric distribution as the model of randomness (Hubert &Arabie, 1985). Let n_{ij} be the number of objects in both class u_i and cluster v_j . Let $n_{i.}$ and $n_{.j}$ be the number of objects in class u_i and cluster v_j respectively. The ARI can be described in Eq. (15). A high value of ARI means a good clustering result.

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i.}}{2} + \sum_j \binom{n_{.j}}{2}\right] - \left[\sum_i \binom{n_{i.}}{2}\sum_j \binom{n_{.j}}{2}\right]/\binom{n}{2}} \tag{15}$$

- Calinski-Harabasz Index (CH) based on cohesion and separation of points and centroids (Calinski & Harabasz, 1974). For a set of N data points $X = (x_1, \ldots, x_N)$ assigned to K clusters $C = \{C_1, \ldots, C_K\}$ with centroids m_i, with $i = 1, \ldots, K$, the CH index is defined in Eq. (16).

$$CH(K) = \frac{T_r(S_B)}{K-1} \Big/ \frac{T_r(S_W)}{N-K} \tag{16}$$

where $T_r(S_B) = \sum_{i=1}^{K} N_i \|m_i - m\|^2$ and $T_r(S_W) = \sum_{i=1}^{K}\sum_{j=1}^{N_i} \left\|x_j - m_i\right\|^2$ are the traces of the between and within-cluster scatter matrices respectively. Here, N_i represents the number of data points belonging to cluster C_i , m is the total mean vector for the entire data set. A large value of $CH(K)$ indicates a clustering result with good quality.

- Davies-Bouldin Index (DB) to maximize the between-cluster distance while minimizing the distance between the cluster centroid and the other data points (Davies &Bouldin, 1979). The DB index is defined as:

$$DB(K) = \frac{1}{K}\sum_{i=1}^{K} R_i \tag{17}$$

where R_i indicates the maximum comparison between cluster C_i and other clusters in the partition, and is written as

$$R_i = max\left(\frac{e_i + e_j}{D_{ij}}\right) \tag{18}$$

where $D_{ij} = \|m_i - m_j\|^2$ is the distance between the centroids of clusters C_i and C_j , e_i and e_j are the average errors for clusters C_i and C_j respectively, and given by $e_i = (1/N_i)\sum_{x \in C_i}\|x - m_i\|^2$. A small value of $DB(K)$ suggests the good clustering results.The experimental results are averages of 20 runs of simulation. The algorithms are implemented with MATLAB R2012a environment on window 7 operating system using Intel Core i3 CPU 3.06 GHz with 4 GB RAM computer. Table 1 summarizes the data sets used in the experiment. Table 2, 3, 4, 5, 6, and 7 present the summary of results as well as centers obtained by various clustering algorithms on different data sets.

Table 1
Description of data sets

Name of data set	Classes	Features	Size of data set
Iris	3	4	150 (50,50,50)
Glass	6	9	214 (70,17,76,13,9,29)
Lung cancer	3	56	32 (9,13,10)
Soyabean(small)	4	35	47 (10,10,10,17)
Wine	3	13	178 (59,71,48)
Vowel	6	3	871 (72,89,172,151,207,180)

Table 2 summarizes the results of KM, ABC and MABCKM algorithms for the six data sets. The OFV values reported are best, average and worst values with standard deviations to indicate the range of values that the algorithms span from the 20 simulations. For iris data set, the proposed algorithm generates excellent results i.e. the worst OFV 78.8518 is even better than best OFV 78.8557 of KM and 79.3444 of ABC. Moreover, its SD is very low i.e. 0.00004 and F-measure 0.8917 is also better than other two algorithms besides having less runtime than ABC. In case of glass data set, the average OFV 336.6181 of proposed algorithm is better than best OFV with much lower SD 0.1604 in comparison to other algorithms. The modified algorithm exhibits comparable F-measure 0.7391 and runtime less than ABC. The hybrid algorithm produces good clusters for lung cancer in terms of best OFV 535.6552 and significant F-measure 0.6648. The average OFV, worst OFV, and SD provided by modified algorithm are much better than ABC but worse than KM. The MABCKM takes less time to execute as compared to ABC. In soyabean data set, average OFV 227.6273 of hybrid algorithm is much better than best OFV of KM and ABC i.e. 232.7593 and 303.3672 respectively. The algorithm provides significant F-measure 0.8134 in addition to lower SD and runtime as compared to ABC. The modified algorithm outperforms others in terms of objective function values and standard deviation for wine data set. The algorithm generates comparable F-measure value 0.7286 and takes less runtime in comparison to ABC. The results for vowel data set also prove the hybrid algorithm in terms of much better objective function values with lowest standard deviation. The F-measure value 0.6034 of MABCKM is not good enough, but runtime is less than that of ABC. It follows that MABCKM is efficient in finding the global optimum with much lower standard deviation.

Tables 3, 4, 5 and 6 provide the best centroids found by the hybrid algorithm in iris, glass, wine and vowel data sets respectively. Table 7 provides the results obtained by various algorithms on different data sets. For iris data set, the modified algorithm generates best values of Sil, ARI, CH and DB indices. In case of glass data set, the CH andDB values provided by hybrid algorithm are best, but the Sil and ARI values are not good enough. The results also prove that the increase in number of dimensions does not affect the behavior of modified algorithm in terms of Sil, ARI, CH and DB values as reported in case of lung cancer data set. Although, the results generated by proposed algorithm are not best in soyabean data set, they are comparable to other algorithms. The results produced by hybrid algorithm are almost identical to that of other algorithms in case of wine data set. With the increase in number of samples, the modified algorithm does not provide best values for all parameters as given in vowel data set, rather the results are best or nearly best in terms of CH and DB values.

From Table 2 and 7, it may be concluded that the proposed algorithm provides best performance on various evaluation measures for iris, glass, lung cancer, and wine data sets, whereas it provides nearly best results for soyabean (small) and vowel data sets. The modified algorithm produces good clustering partitions on low-dimensional as well as high-dimensional data sets. The MABCKM algorithm takes less runtime in comparison to ABC on all data sets.

Table 2
Results obtained by various algorithms on different data sets. Bold face indicates the best and italic face the second best result

Data sets	Best OFV	Average OFV	Worst OFV	Standard deviation (SD)	F-measure	Runtime (sec)
Iris (k=3)						
KM	*78.8557*	108.1461	146.3252	11.1602	0.8852	0.7720
ABC	79.3444	*83.7317*	*119.6221*	*8.1109*	0.8852	12.0355
MABCKM	**78.8514**	**78.8516**	**78.8518**	**0.00004**	**0.8917**	10.6493
Glass (k=6)						
KM	*336.2926*	*377.2512*	501.7372	*24.8781*	**0.7465**	2.5525
ABC	398.1191	452.7072	*498.9743*	30.5707	0.7335	15.8255
MABCKM	**336.0840**	**336.6181**	**337.6870**	**0.1604**	*0.7391*	14.8477
Lung cancer (k=3)						
KM	*544.7792*	**548.9163**	**557.2380**	**1.3072**	*0.6350*	2.4236
ABC	739.2245	765.7712	803.0120	21.3254	0.5646	4.7072
MABCKM	**535.6552**	*570.8732*	*641.3114*	*10.5690*	**0.6648**	4.6313
Soyabean (k=4)						
KM	*232.7593*	*249.7512*	**290.9811**	**6.2142**	0.7308	2.4035
ABC	303.3672	343.5312	368.00	20.9647	*0.7854*	4.7927
MABCKM	**208.1545**	**227.6273**	*309.7732*	*10.2121*	**0.8134**	4.6968
Wine (k=3)						
KM	**2.3707e6**	2.4818e6	2.9145e6	*55022.31*	0.7286	2.6385
ABC	2.3872e6	*2.4387e6*	2.6092e6	71422.42	**0.7502**	28.3035
MABCKM	**2.3707e6**	**2.3708e6**	**2.3710e6**	**29.9780**	0.7286	26.1589
Vowel (k=6)						
KM	3.1445e7	3.4203e7	5.1395e7	*2.5791e6*	**0.6467**	2.6898
ABC	*3.1167e7*	*3.3984e7*	*4.2796e7*	3.2554e6	*0.6083*	115.0184
MABCKM	**3.06907e7**	**3.06907e7**	**3.06908e7**	**10.1211**	0.6034	102.2788

Table 3
Centers obtained for the best OFV on iris data set

Center 1	5.9016	2.7484	4.3935	1.4339
Center 2	6.8500	3.0737	5.7421	2.0711
Center 3	5.0060	3.4280	1.4620	0.2460

Table 4
Centers obtained for the best OFV on glass data set

Center1	1.5138	13.3433	0.8933	3.1867	70.3567	4.7000	6.5867	0.7333	0
Center2	1.5173	13.1319	3.4992	1.3818	72.8125	0.5844	8.3484	0.0277	0.0642
Center3	1.5201	13.1335	0.5729	1.4865	73.0682	0.5018	11.0053	0.0141	0.0618
Center4	1.5283	11.8671	0	1.2186	71.6729	0.2514	14.3157	0.4500	0.1371
Center5	1.5213	13.8856	3.3436	1.0492	71.7956	0.1969	9.5303	0.0764	0.0492
Center6	1.5163	14.6746	0.1654	2.1292	73.3138	0.0708	8.5804	0.9869	0.0150

Table 5
Centers obtained for the best OFV on wine data set

Center1	12.9	2.5	2.4	19.9	103.6	2.1	1.6	0.4	1.5	5.7	0.9	2.4	728.3
Center2	13.8	1.9	2.4	17.0	105.5	2.9	3.0	0.3	1.9	5.7	1.1	3.1	1195.1
Center3	12.5	2.5	2.3	20.8	92.3	2.1	1.8	0.4	1.5	4.1	0.9	2.5	458.2

Table 6
Centers obtained for the best OFV on vowel data set

Center 1	402.1	1026.1	2333.6
Center 2	624.2	1314.6	2341.1
Center 3	519.8	1791.7	2540.8
Center 4	412.5	2103.4	2651.4
Center 5	449.4	995.5	2674.8
Center 6	368.4	2292.3	2968.5

Table 7
Results obtained by various algorithms for different data sets. Bold face indicates the best and italic face the second best result

Data set	Silhouette (Sil)	Adjusted Rand Index (ARI)	Calinski-Harabasz Index (CH)	Davies-Bouldin Index (DB)
Iris (k=3)				
KM	0.7344	0.7163	561.5941	0.5900
ABC	0.7344	0.7163	561.5941	0.5900
MABCKM	**0.7357**	**0.7302**	**561.6282**	**0.5253**
Glass (k=6)				
KM	*0.6213*	**0.2738**	*124.5012*	*0.8472*
ABC	**0.6246**	*0.2691*	123.2140	0.8510
MABCKM	0.6123	0.2589	**124.6052**	**0.7567**
Lung cancer (k=3)				
KM	*0.0784*	*0.1884*	*2.5785*	2.7193
ABC	0.0682	0.1016	2.3693	*1.8722*
MABCKM	**0.1453**	**0.2062**	**2.8694**	**1.3862**
Soyabean (k=4)				
KM	**0.5907**	**0.5924**	28.1674	**0.9689**
ABC	*0.4964*	0.5451	**33.5753**	1.1079
MABCKM	0.4915	*0.5634*	*33.1912*	*1.0058*
Wine (k=3)				
KM	0.7323	0.3711	561.816	**0.4234**
ABC	0.7323	0.3711	561.816	0.5210
MABCKM	0.7323	0.3711	561.816	**0.4234**
Vowel (k=6)				
KM	**0.5514**	**0.3823**	1426.33	0.7663
ABC	0.5236	*0.3158*	*1462.17*	**0.7557**
MABCKM	*0.5251*	0.3122	**1465.64**	*0.7638*

By increasing iteration from 1 to 100, Fig. 1 to 6 describe the change in objective function value by using different methods. From Fig. 1 to 6, it may be summarized as:

- The KM method provides stable and better convergence as compared to ABC. It converges to the global optimum or near global optimum for all data sets except soyabean (small).
- The ABC method depicts slow and less stable performance as compared to KM. However, it performs better and converges to the global optima or near global optima in iris, wine and vowel data sets. The results prove that ABC is not able to generate sufficiently good results on high dimensional data sets.
- The hybrid method shows fast, stable and best performance for all data sets. The algorithm converges to the global optima every time and provides excellent results irrespective of number of samples or dimensions of data sets.

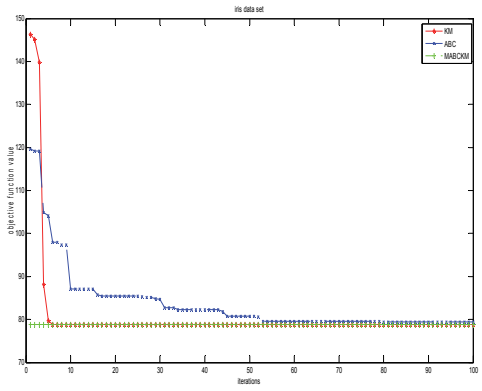

Fig.1. Comparison of OFV on iris data set

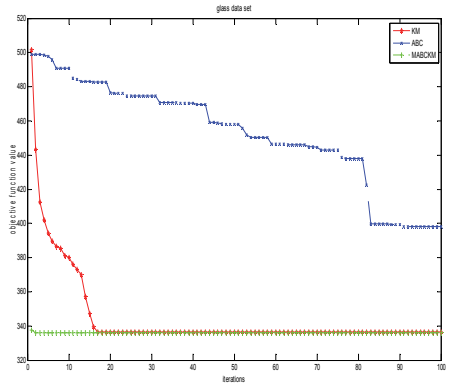

Fig. 2. Comparison of OFV on glass data set

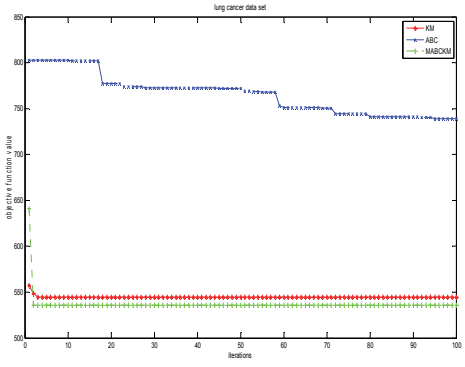

Fig. 3. Comparison of OFV on lung cancer data set

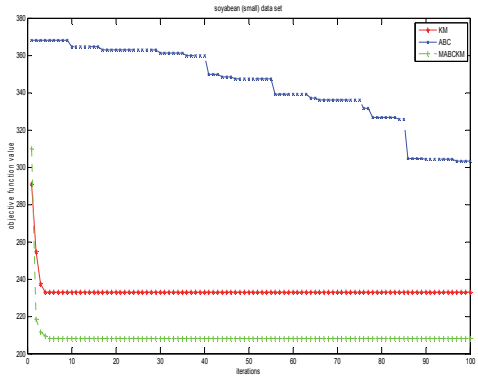

Fig. 4. Comparison of OFV on soyabean (small) data set

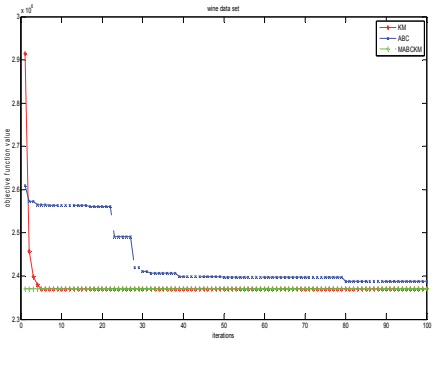

Fig. 5. Comparison of OFV on wine data set

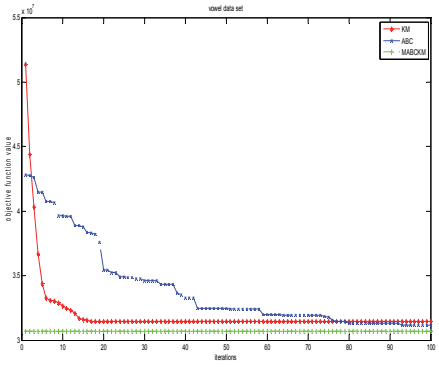

Fig. 6. Comparison of OFV on vowel data set

6. Conclusion

This paper has presented a hybrid clustering algorithm (MABCKM) based on modified ABC and KM algorithms. The proposed method exhibits the qualities of both the algorithms. The modified ABC algorithm incorporates modified initialization phase to generate better initial solutions. Moreover, it makes use of variable tournament selection in place of roulette wheel selection in onlooker bee phase, so as to provide better exploration and exploitation of solution space in addition to enhanced convergence speed. The performance of the algorithm is evaluated in terms of different parameters on six standard data sets from UCI Machine Learning Repository and compared with ABC and KM algorithms. The experimental results show that the proposed MABCKM algorithm is able to escape local optima and find better objective function values with much lower standard deviation in comparison to other two algorithms. The proposed algorithm also outperforms the other methods in terms of the F-measure, silhouette, ARI, CH and DB indices and achieves best ranking among three methods. The results prove that the modified algorithm produces better clustering partitions and leads naturally to the conclusion that MABCKM is a viable and robust technique for data clustering. The

proposed method needs improvement to perform automatic clustering without any prior knowledge of number of clusters.

References

Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications*, *37*(8), 5682-5687.

Al-Sultan, K.S. (1995). A tabu search approach to the clustering problem. *Pattern Recognition, 28*(9), 1443-1451.

Blickle, T., & Thiele, L. (1995). *A mathematical analysis of tournament selection,* in: Eshelman, L.(Ed.) Proceedings of Sixth International Conf. Genetic Algorithms (ICGA95), Morgan Kaufmann, San Francisco, CA, pp. 9-16.

Calinski, R., &Harabasz, J. (1974). A dendrite method for cluster analysis. *Commun. Stat., 3*(1), 1-27.

Chen, C.Y., & Ye, F. (2004). Particle swarm optimization algorithm and its application to clustering analysis. In *IEEE International Conference on Networking, Sensing and Control, Taiwan*, pp. 789-794.

Dalli, A. (2003). Adaptation of the F-measure to cluster-based Lexicon quality evaluation. In *Proceedings of EACL 2003 Workshop, Budapest,* pp. 51-56.

Davies, D. L., &Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2), 224-227.

Gan, G., Ma, C., & Wu, J. (2007). *Data clustering: Theory, algorithms, and applications*. ASA-SIAM Series on Statistics and Applied Probability, SIAM, Philadelphia, VA.

Gao, W., & Liu, S. (2012). A modified artificial bee colony algorithm. *Computers & Operations Research, 39*(3), 687-697.

Handl, J., Knowles, J., &Dorigo, M. (2003). On the performance of ant-based clustering. *Design and Application of Hybrid Intelligent Systems: Frontiers in Artificial Intelligence and Applications,104*, 204-213.

Hubert, L., &Arabie, P. (1985). Comparing Partitions. *Journal of Classification, 2*, 193-218.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report – TR06,* Erciyes University.

Karaboga, D., &Ozturk, C. (2011). A novel clustering approach: Artificial bee colony (abc) algorithm. *Applied Soft Computing, 11*(1), 652-657.

Kumar, Y., &Sahoo, G. (2014). A charged system search approach for data clustering. *Progress in Artificial Intelligence, 2*, 153-166.

Miller, B.L., & Goldberg, D.E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems, 9*, 193-212.

Murthy, C.A., & Chowdhury, N. (1996). In search of optimal clusters using genetic algorithms. *Pattern Recognition Letters, 17*(8), 825-832.

Rahnamayan, S., Tizhoosh, H.R., &Salama, M.M.A. (2008). Opposition-based differential evolution. *IEEE Transactions on Evolutionary Computation,12*(1), 64–79.

Santosa, B., &Mirsa, K.N. (2009). Cat swarm optimization for clustering. In *International Conference on Soft Computing and Pattern Recognition (SOCPAR'09)*, pp. 54-59.

Satapathy, S.C., &Naik, A. (2011). Data clustering based on teaching-learning-based optimization. In *Swarm, Evolutionary, and Memetic Computing*, Springer Berlin Heidelberg, pp. 148-156.

Selim, S.Z., &Alsultan, K. (1991). A simulated annealing algorithm for the clustering problem. *Pattern Recognition, 24*(10), 1003-1008.

Shelokar, P.S., Jayaraman, V.K., & Kulkarni, B.D. (2004). An ant colony approach for clustering. *AnalyticaChimicaActa, 509*(2), 187-195.

Xu, R., Xu, J., &Wunsch II, D.C. (2012). A comparison study of validity indices on swarm-intelligence-based clustering. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 42*(4), 1243-1256.