Performance Analysis of Data Clustering Algorithms using Various Effectiveness Measures

Krishnamoorthi Murugasamy and Natarajan Mathaiyan
Department of Computer Science and Engineering, Bannari Amman Institute of Technology, India

Abstract: *Data clustering is a method to group the data records that are similar to each other. In recent days, researcher show significant attention towards the use of swarm based optimization algorithms to improve the performance of clustering process. This Performance analysis concentrates on the effectiveness of five different algorithms with respect to various distances metrics to find the effective algorithm among them. The algorithms used for comparison are K-means algorithm, Artificial Bee Colony (ABC) algorithm, Fuzzy C-Means (FCM) incorporated ABC (ABFCM) algorithm, K-means incorporated Artificial Bee Colony (ABK) algorithm and Bacterial Foraging Optimization algorithm (BFO). Among those algorithms, ABFCM and ABK algorithms are enhanced ABC algorithm in which the FCM and K-means operator are incorporated in the sc out phase of the traditional ABC algorithm respectively. In this paper, the performance of these algorithms are compared in terms of various distances metrics like dice coefficient, jaccard coefficient, beta index and distance index by varying the cluster sizes and number of iteration. Finally, from the experimental results it proves that the proposed algorithms ABFCM and ABK outperforms better when compared with the existing algorithms.*

Keywords: *Data clustering, k-means algorithm, FCM, ABC, distances metrics.*

Received November 5, 2012; accepted February 24, 2013; published online June 11, 2015

1. Introduction

A data mining technique which is most commonly used in many research fields is clustering. Some of the research fields which use the clustering techniques are statistical pattern recognition, information recovery, machine learning and data mining [4]. Clustering is the technique that deals with the unsupervised division of patterns into clusters. Some of the clustering approaches are Partitioning, Hierarchical, Density based clustering, Fuzzy clustering, artificial neural clustering [2, 3, 5, 6, 12, 15, 16], Statistical clustering and Grid based clustering. The partitioning and hierarchical clustering approaches are the two fundamental approaches among the different clustering approaches which are being used in research areas [8]. The partitioning clustering method partition the dataset into predefined number of clusters whereas the hierarchical clustering method iteratively split the dataset into smaller subsets, until some termination condition is satisfied. The most commonly used clustering method is partitioning clustering method due to use of less memory and time of execution.

The most popular class of partitional clustering algorithms is K-means algorithm [11], which is simple, fast and center based algorithm. But K-means algorithm highly depends on the initial cluster center and always converges to the nearest local optimum from the starting position of the search. For clustering the data, Fuzzy C-Means (FCM) [2, 7] technique has been used from the early days. In clustering and classification [1, 10, 14, 18, 19, 21], the fuzzy clustering is widely applied and is successful too. Many researches are carried out using diverse techniques for data clustering. Mualik and Bandyopadhyay [13] have suggested a technique using genetic algorithm to decide the clustering issue which was experimented on synthetic and real life datasets to calculate the performance. Krishna and Murty [11] have suggested a model called genetic K-means algorithm for clustering examination which expresses a vital mutation operator controlled clustering known as distance-based mutation.

The major difficulty in the clustering algorithms is that they have no optimization functions for optimizing the clusters. Optimization is a necessary process that makes the clustering efficient when the redundant data collections are considered. For cluster optimization, various optimization algorithms such as genetic algorithm, particle swarm optimization, ant colony algorithm and tabu search arrived later. Recently, Artificial Bee Colony (ABC) algorithm for cluster optimization is suggested by Karaboga and Ozturk [9]. The ABC algorithm works based on the behavior of the honey bees which search for food. Honey bees are one of the most closely studied social insect. Their behavior on food search, knowledge, remembering and information sharing features are some of the most interesting study areas in swarm intelligence [17]. To enhance the performance of the ABC algorithm in clustering, many algorithms were proposed by combining with certain features.

In this paper, the performance of different data clustering techniques is compared using various distance metrics. The purpose of this work is to

analyze the performance of five different clustering algorithms that are taken the standard clustering problem to find the effective solution. The techniques used for our comparison are K-means algorithm, ABC algorithm, FCM incorporated ABC algorithm, K-means incorporated ABC algorithm and Bacterial Foraging Optimization (BFO) algorithm. In FCM incorporated ABC algorithm and K-means incorporated ABC algorithm, the FCM functions and the K-means functions are incorporated at the scout bee phase of the ABC algorithm. Because the third stage of the ABC algorithm is a random phase which would be used when an abandon solution is generated. The performance of the algorithms are evaluated and compared in terms of dice coefficient, jaccard coefficient, beta index and distance index with different iterations and different cluster size.

This paper is organized as follows, section 2 elaborates the techniques which we used for our comparison and section 3 describes the terms considered for our comparison and section 4 gives a brief idea about the data set used and section 5 narrates the performance analysis of different algorithms and finally section 6 concludes our comparison.

2. Techniques Used for Comparison

The techniques used for our comparison are K-means algorithm, ABC algorithm, FCM incorporated ABC algorithm, K-means incorporated ABC algorithm and BFO algorithm. The brief explanations of those techniques are as follows.

2.1. BFO-based Clustering Algorithm

The BFO [20] is an algorithm which uses the searching model based on the foraging behavior of E.Coli bacteria which is present in the human intestine. The bacterial foraging technique is used to solve the non-gradient optimization issue. The BF technique uses three processes to solve the non-gradient optimization problem; they are chemo taxis, reproduction and elimination and dispersal. Usually, the E.coli bacteria tries to discover food and avert the harmful phenomena while foraging and after some time delay it would recover and return to some standard behavior in a homogeneous medium. The E.Coli bacterium can move in two diverse ways; they are tumbling and swimming. In its whole lifetime, the E.Coli bacterium alternates amid those two modes of operation. The alteration amid those two modes is called chemo tactic steps which will move the bacterium in random direction and enables it to pursuit for food. When the bacterium collects the necessary amount of food, it would reproduce by dividing into two. The population of the bacteria would also get altered by the local environment.

The clustering tasks are pondered as the optimization problem. The fitness function should be mentioned first. Here, we take the sum of squared error as the required function G in bacterial foraging based clustering.

$$G(w,z)=\sum_{b=1}^{L}\sum_{r=1}^{m}\sum_{d=1}^{D} w_{rp}\left\|x_{rd}-z_{bd}\right\|^2 \quad (1)$$

Where D: Dimension of the search space, w: Weight matrix of size $m\times L$, and w_{rp}: Associated weight of data x_r with cluster b.

$$w_{rp}=\begin{cases}1 & \text{if } x_r \text{ is labelled to cluster } b, \text{ where } r=1,...,m;\ b=1,...,L\\ 0 & \text{otherwise}\end{cases} \quad (2)$$

In bacterial foraging based clustering algorithm, S-size population of the bacteria is created for each center, such that there will be $S\times L$ bacteria altering positions for minimum cost by foraging behaviors in this technique. At the beginning, the S data is generated randomly from X as bacteria for each center z_b. The chemo taxis process starts for every bacterium i. The entire bacteria update their positions for N_b step of iterations. The agents first present a tumble in a unit length random direction with a fundamental chemo taxis step size and then swim to minimize the objective function G up to maximum number of steps N_s. After the iteration process, the entire bacteria will get converged to particular places in the search space. The eventual positions of the bacteria are pondered as the required centers.

2.2. ABC-based Clustering Algorithm

ABC is an algorithm which is explained by Karaboga in 2005, inspired by the smart behavior of honey bees. Karaboga and Ozturk [9] have used ABC algorithm for data clustering. The colony of artificial bees has three set of bees namely employed, onlookers and scouts bees. A bee which is waiting on the dance area for making a choice to pick a food source is called onlooker and the bee which goes to the food source that is selected by the onlooker is called employed bee. The other type of bee is scout bee that carries out unsystematic search for discovering novel sources. The position of a food source denotes a realistic solution to the optimization issue and the nectar value of a food source related to the quality (fitness) of the associated solution, estimated by:

$$FIT_i=\frac{1}{1+f_i} \quad (3)$$

The bees which have the fitness values as good enough is the result of this fitness. The detailed pseudo code of the ABC algorithm (Algorithm 1) is given below.

Algorithm 1: ABC.

Initialize the population of the solutions $X_{i,j}$
Evaluate the population.
Setcycle=1, *cycle represents an iterative value.*
Produce new solutions (food source positions) $v_{i,j}$ *in the neighborhood of* $X_{i,j}$, *using the formula,*

$$v_{i,j} = x_{i,j} + \Phi_{i,j}(x_{i,j} - x_{k,j}) \quad (4)$$

Where Φ *is a random number in the range* [-1, 1].
Apply the greedy selection process between $v_{i,j}$ *and* $X_{i,j}$ *based on the fitness*
Calculate the probability values P_i *for the solutions* $X_{i,j}$ *by means of their fitness values using the equation.*

$$p_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (5)$$

In order to calculate the fitness values of solutions we have employed the following equation.

$$fit_i = \begin{cases} if\ f_i \geq 0, & \frac{1}{1+f_i} \\ if\ f_i \leq 0, & 1+abs(f_i) \end{cases} \quad (6)$$

Normalize Pi values into [0, 1].
Produce the new solutions (new positions) $v_{i,j}$ *for the onlookers from the solutions xi, selected depending on Pi, and evaluate them*
Apply the greedy selection process for the onlookers between x_i *and* v_i *based on fitness*
Determine the abandoned solution (*source*), *if exists, and replace it with a new randomly produced solution* x_i *for the scout using the equation.*

$$x_{i,j} = min_j + rand(0,1) \times (max_j - min_j) \quad (7)$$

Memorize the best food source position (*solution*) *achieved so far*
*Cycle=cycle+*1
Until cycle= Maximum Cycle Number (*MCN*).

2.3. ABK Algorithm

ABK algorithm is a hybrid algorithm which incorporates the K-means algorithm into the ABC algorithm. The ABC algorithm is recently introduced into cluster optimization whereas K-means algorithm is most widely used in clustering. In ABC algorithm the employed bee and the onlooker bee phase are mandatory phases in ABC algorithm and the scout bee phase is an unsystematic phase. So, the K-means algorithm is applied in the scout bee phase. The addition of the novel solution from the K-means after every cycle may increase the reach of ABC algorithm to a different level.

The ABC algorithm has several dimensional search spaces in which there are Employed bees and Onlookers bees. Both the bees are categorized by their experience in identifying the food source. The initial population is opted from the employed bee phase and the food location is possessed by this employed bee. The solution of the employed bee is altered in the onlooker bee stage based on the following formula:

$$u_{i,j} = s_{i,j} + \Phi_{i,j}(s_{i,j} - s_{k,j}) \quad (8)$$

Where $S_{i,j}$: Solution obtained from the employed bee phase. $\phi_{i,j}$: Randomly produced number of range [-1, 1], and $S_{k,j}$: Random indexes in the solution matrix of the employed bee.

A novel solution is created based on the formula and the solution is applied in the fitness function to obtain the fitness value. If the new fitness value is better than the old one, then the new fitness value is selected and the old one would get eliminated. This process would last until the entire employed bee gets processed. The scout bee phase is the eventual stage of the ABC algorithm, in which K-means operator is implemented in order to find the food source. The scout bee initiates the process by choosing the solution from the onlooker bee phase which poses the lowest fitness value. The onlooker bee phase generates diverse solution based on different $u_{i,j}$ values. The solution with the least fitness value is selected and the distance matrix is computed. Based on the distance values in the matrix, the data points are grouped with respect to the minimum distance value. Then, the centroid is computed by taking the mean values of the data points in the cluster. Then the computed centroid is given a new set of solution for scout bee phase.

2.4. ABFCM Algorithm

In ABFCM algorithm, the FCM functions are incorporated into the ABC algorithm. In ABC algorithm Employed bee and onlooker bee are inexorable and the scout bee is a random phase. So, the FCM operator is applied in the scout bee to improve the effectiveness of data clustering. The FCM operator generates the solution based on the solutions from the first two stages i.e., employed bee and onlooker bee. The optimization problem may get reduced by giving more suitable solutions using this process. The addition of the new solution may increase the performance of the ABC algorithm.

The function which is mentioned below is iteratively optimized for clustering. The membership function is updated with the following formula for the entire iteration:

$$u_{ij} = \frac{1}{\sum_{k-1}^{c} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \quad (9)$$

The centroid of the cluster is updated by the following formula and this solution is added into the scout bee phase:

$$u_{ij} = \frac{\sum_{i-1}^{N} u_{ij}^{m} . x_i}{\sum_{i-1}^{c} u_{ij}^{m}} \quad (10)$$

Where $u_{i,j}$: Degree of membership d_i in cluster j, and c: Centroid cluster.

2.5. K-Means Algorithm

K-means [12] is an algorithm which is used to divide or to group the objects based on attributes (features) into k number of groups. Here, k is the number which is positive integer. The fundamental procedure of K-

means clustering is uncomplicated. At the first stage, we require establishing the number of clusters *k* and we assume the centroid for these clusters. We can take any object as an initial centroid unsystematically. The K-means algorithm will do the following process until convergence.

- Establish the centroid coordinate.
- Discover the distance matrix of every object with relevant to the centroids.
- Group the object based on short distance.
- Find the new centroid.

3. Parameters Used for Evaluating the Quality of Clustering

The terms which we considered for the comparison of the different techniques are dice coefficient, jaccard coefficient, beta index and distance index. Here, the first two metrics need the original cluster to find the evaluation metric value whereas the last two measures are the used to evaluate the algorithm without the help of original cluster. The explanations of the considered parameter are as follows.

3.1. Dice Coefficient

The formula to calculate the dice coefficient is given below. It is the ratio of the product of numerical integer two and the modulus of integrated values of two clusters to the sum of modulus of the original cluster and the modulus of cluster obtained after applying the algorithm.

$$s = \frac{2|X \cap Y|}{|X| + |Y|} \tag{11}$$

Where *S*: Dice coefficient, *X*: Original cluster, and *Y*: Cluster formed after applying the algorithm.

3.2. Jaccard Coefficient

The jaccard coefficient is defined as the ratio of size of intersection to the size of union of the sample sets (clusters). The formula for calculating the jaccard coefficient is as follows:

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{12}$$

Where *J*: Jaccard coefficient, *X*: Original cluster, and *Y*: Cluster formed after applying the algorithm.

3.3. Beta Index

The beta index is the ratio of total variation to the variation within the class and it is expressed as equation as follows:

$$\beta = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n_i}(X_{ij} - \overline{X})^2}{\sum_{i=1}^{c}\sum_{j=1}^{n_i}(X_{ij} - \overline{X_i})^2} \tag{13}$$

Where β: Beta index, $\overline{X}$: Mean of all the data points, $\overline{X_i}$: Mean of the data points that belongs to cluster c_i, X_{ij}: j^{th} data point of i^{th} cluster, and n_i: Number of data points in cluster c_i

3.4. Distance Index

The distance index is defined as the ratio of average intra-cluster distance to the average inter-cluster distance. The formula for calculating the distance index is as follows:

$$Dis = \frac{Intra}{Inter} \tag{14}$$

The Intra value is calculated using the equation given below:

$$Intra = \frac{1}{n}\sum_{i=1}^{K}\sum_{x_j \in c_i}\left\|x_j - z_i\right\|^2 \tag{15}$$

The value of Inter is calculated using the following formula:

$$Inter = \frac{1}{K}\sum\left\|z_i - z_j\right\|^2 \tag{16}$$

Where $i = 1, 2, \ldots, k\text{-}1$ and $j = i+1, \ldots, k$.

4. Dataset Description

The datasets which we taken for our comparison are from the UCI machine learning repository. We have chosen six datasets for our comparison of five different techniques. The data sets which we used in our comparison are abalone, zoo, iris, wine, thyroid disease and liver disorder. These datasets are considered for comparison and the results for those different techniques are compared with each other.

4.1. Abalone Dataset

The attributes of this dataset are categorical, integer and real. The number of instances in this dataset is 4177 and the number of attributes is 8 and this dataset has no missing values.

4.2. Zoo Dataset

The attribute of this dataset are categorical and integer. The number of instances in this dataset is 101 and the number of attributes is 17 and this dataset has no missing values.

4.3. Iris Dataset

The dataset attribute characteristic are real. The number of instances in this dataset is 150 and the number of attributes is 4 and it has no missing values.

4.4. Wine Dataset

The dataset attribute characteristic are integer and real and the number of instances of this dataset is 178 and

the number of attributes is 13 and it has no missing values.

4.5. Liver Disorders Dataset

The dataset attribute characteristics of this dataset are categorical, integer and real. The number of instances of this dataset is 345 and the number of attributes is 7 and it has no missing values.

4.6. Thyroid Disease Dataset

The attribute characteristics of this dataset are categorical and real. The number of instances is 7200 and the number of attributes is 21.

5. Performance Analysis

This section explains the performance of various algorithms chosen for comparison in terms of Dice Coefficient, Jaccard Coefficient, Beta Index and Distance Index with different iterations and different cluster sizes using the datasets taken from the UCI machine learning repository.

5.1. Performance based on Cluster Variation

This section shows the performance of the techniques used for comparison based on the cluster variation. The comparison is performed in this section using four and five clusters. The values in the table with best performance are shown in bold text.

5.1.1. Performance based on Four Clusters

Table 1 shows the values obtained for the algorithms of cluster size four after fixing the iteration as five. Here, the comparison was done using the parameters discussed in section 3. In this Table 1, the performance of all the algorithms shows similar performance in terms of dice and jaccard coefficient using the dataset abalone. In terms of beta index and distance index, the AB-FCM algorithms performed well using the dataset Abalone. The performance of ABK and ABFCM is better in terms of dice coefficient using the dataset Zoo. Using the dataset iris, the ABFCM shows better performance contrast to other algorithms in terms of dice coefficient, jaccard coefficient and distance index.

While in the Wine dataset, the algorithms show same performance in terms of dice and jaccard coefficient. In terms of distance index, the ABFCM algorithm shows better performance compared to other algorithms using the dataset Wine. Using the Liver dataset, the performance of ABK algorithm is better when compared to other algorithms in terms of distance index. The performance of ABFCM is better using the dataset liver in terms of beta index. In thyroid dataset, the ABK algorithm performed better compared to other algorithms in terms of beta index and the ABFCM algorithm performed well in terms of distance index.

Table 1. Performance comparison based on five iterations and four clusters.

Cluster=4, Iteration=5							
		Abalone	**Zoo**	**Iris**	**Wine**	**Liver**	**Thyroid**
K-Means	**Dice**	2.0	0.6	1.901477	2.0	**1.958974**	2.0
	Jaccard	1.0	0.30	0.950738	1.0	0.979487	1.0
	Beta	0.040083	**0.717457**	0.212934	**0.383948**	0.804191	0.172037
	Distance	0.009244	**0.007378**	0.013138	0.015645	0.052525	0.011191
BFO	Dice	2.0	0.653721	1.900621	2.0	1.948051	2.0
	Jaccard	1.0	0.326860	0.950310	1.0	0.974025	1.0
	Beta	0.059906	0.621098	0.059757	0.0612	0.061474	0.061314
	Distance	0.017410	0.028525	0.022202	0.015280	0.020017	0.018415
ABC	Dice	2.0	0.740229	1.870370	2.0	1.952662	2.0
	Jaccard	1.0	0.370114	0.935185	1.0	0.976331	1.0
	Beta	0.232303	0.341884	**0.473487**	0.312005	0.161657	0.584262
	Distance	0.014576	0.012321	0.037750	0.055719	0.014980	0.030387
AB-K	Dice	**2.0**	**0.740229**	1.869767	**2.0**	1.944260	**2.0**
	Jaccard	**1.0**	0.370114	0.934883	**1.0**	0.972130	**1.0**
	Beta	0.224750	0.341884	0.189268	0.011532	0.808726	**0.886186**
	Distance	0.014808	0.012321	0.012249	0.005618	**0.005078**	0.173132
AB-FCM	Dice	**2.0**	**0.740229**	**1.922779**	**2.0**	1.944250	**2.0**
	Jaccard	**1.0**	0.370114	**0.961389**	**1.0**	0.972125	**1.0**
	Beta	**0.233170**	0.341884	0.298432	1.115322	**0.899726**	0.437694
	Distance	**0.006369**	0.012321	**0.006914**	**0.005618**	0.006778	**0.010258**

5.1.2. Performance based on Five Clusters

Table 2 shows the values obtained for the different techniques in terms of dice coefficient, jaccard coefficient, beta index and distance index with five iterations and for cluster sizes of five using different datasets. In abalone dataset in terms of dice and jaccard coefficient, all the algorithms show best and similar performance. In terms of distance index, the ABFCM algorithm shows best performance compared to the other algorithms which we used for comparison.

When considering the zoo dataset for our comparison in terms of Dice Coefficient, the algorithms ABK and ABFCM shows better performance compared to the other algorithms. In terms of Jaccard Coefficient using the dataset Zoo, the techniques ABK and ABFCM performed well compared to the K-means and ABC. For the Iris dataset when comparing the techniques in terms of dice coefficient, the ABFCM performed better compared to other algorithms. In terms of jaccard coefficient for the dataset Iris, the performance of the ABFCM is better than the other algorithms.

Table 2. Performance comparison based on five iterations and five clusters.

Cluster=5, Iteration=5							
		Abalone	**Zoo**	**Iris**	**Wine**	**Liver**	**Thyroid**
K-Means	**Dice**	2.0	0.5485232	1.9148936	2.0	**1.9550561**	2.0
	Jaccard	1.0	0.2742616	0.9574468	1.0	**0.9775280**	1.0
	Beta	0.1988337	0.2041679	0.1606912	**0.2975077**	0.0181374	0.1623388
	Distance	0.0108735	0.0137580	0.0145674	0.0227009	0.0119341	0.0123087
BFO	Dice	2.0	0.6537216	1.9006211	2.0	1.9480519	2.0
	Jaccard	1.0	0.3268608	0.9503105	1.0	0.9740254	1.0
	Beta	0.0599068	**0.0621098**	0.0597576	0.0612	0.0614742	0.0944320
	Distance	0.0174102	0.0285255	0.0222202	0.0152804	0.0200175	0.1226358
ABC	Dice	2.0	0.7402298	1.8703703	2.0	1.9526627	2.0
	Jaccard	1.0	0.3701149	0.9351851	1.0	0.9763313	1.0
	Beta	**0.3222622**	0.3418844	0.2734874	0.2578448	0.0616571	1.3367803
	Distance	0.0187980	**0.0123217**	0.0377503	0.0301477	0.0149805	1.2735795
ABK	Dice	**2.0**	**0.7402298**	1.8733031	**2.0**	1.9442508	**2.0**
	Jaccard	**1.0**	**0.3701149**	0.9366515	**1.0**	0.9721254	**1.0**
	Beta	0.3041679	0.3418844	0.0836999	0.0707185	**0.0809726**	**0.4502965**
	Distance	0.0192641	0.1232170	0.0097454	**0.0056354**	**0.0067787**	0.0279585
ABFCM	Dice	**2.0**	**0.7402298**	**1.9227799**	**2.0**	1.9442508	**2.0**
	Jaccard	**1.0**	**0.3701149**	**0.9613899**	**1.0**	0.9721254	**1.0**
	Beta	0.129683	0.3418844	**0.2984326**	0.0707185	**0.0809726**	0.3722272
	Distance	**0.0062490**	0.1232170	**0.0069146**	**0.0056354**	**0.0067787**	**0.0088173**

In terms of beta index for the iris dataset, the AB-FCM algorithm shows better performance compared to other algorithms. In terms of Distance Index for the dataset Iris, the performance of AB-FCM algorithm is better compared to other algorithms. While considering the wine dataset for the comparison in terms of dice and jaccard coefficient, all the algorithms shows similar performance. In terms of distance index for wine, the algorithms ABK and ABFCM shows better compared to other techniques. When comparing the Liver dataset in terms of Beta Index, the algorithms ABK and ABFCM performed well compared to the other algorithms. In terms of distance index for the dataset liver, the ABK and ABFCM shows better performance than other algorithms. When comparing the dataset Thyroid for the performance of different algorithms in terms of dice and jaccard coefficient, all the algorithms show similar performance. In terms of distance index for the thyroid dataset, ABFCM algorithm shows better performance compared to other algorithms.

5.2. Performance based on Iteration Variation

This section explains the performance of the techniques we used for comparison by varying the iterations. The performances are compared using ten iterations and twenty iterations.

5.2.1. Performance based on Ten Iterations

Table 3 details the performance of the techniques used for comparison with ten iterations and cluster sizes of three. The abalone dataset in terms of dice and jaccard coefficient, all the algorithms shows similar performance. The ABK algorithm shows better performance in terms of Beta Index compared to other algorithms using the dataset Abalone. In terms of distance index, ABFCM algorithm shows better performance compared to other algorithms using the dataset Abalone. While comparing the performance using the dataset zoo in terms of jaccard coefficient and beta index, ABFCM shows similar performance than other algorithms. Using the iris dataset when comparing the performances of algorithms in terms of dice coefficient, jaccard coefficient and distance index, ABFCM algorithm shows better performance compared to other algorithms. When comparing the performance using the dataset wine in terms of dice and jaccard coefficient, all the algorithms show similar performance. Using the dataset wine in terms distance index, ABK and ABFCM gives best performance than other algorithms. When comparing the performances of the different algorithms using the dataset thyroid in terms of dice coefficient and jaccard coefficient, all the algorithms shows similar performance. Using the thyroid dataset in terms of distance index, the ABK algorithm shows best performance than other algorithms.

Table 3. Performance comparison based on ten iterations and three clusters.

Iteration=10, Cluster=3							
		Abalone	**Zoo**	**Iris**	**Wine**	**Liver**	**Thyroid**
K-Means	Dice	2.0	0.604477	1.920398	2.0	1.950617	2.0
	Jaccard	1.0	0.302238	0.960199	1.0	0.975308	1.0
	Beta	0.106478	0.141228	0.342244	**0.236636**	0.267324	0.371288
	Distance	0.008644	**0.007446**	0.010625	0.016195	0.008508	0.011025
BFO	Dice	2.0	0.653721	1.900621	2.0	1.948051	2.0
	Jaccard	1.0	0.326860	0.950310	1.0	0.974025	1.0
	Beta	0.059906	0.062109	0.059757	0.0612	0.061474	0.059906
	Distance	0.017410	0.028525	0.022202	0.015280	0.020017	0.017410
ABC	Dice	2.0	**0.740229**	1.870370	2.0	**1.951515**	2.0
	Jaccard	1.0	**0.370114**	0.935185	1.0	**0.975757**	1.0
	Beta	0.239303	**0.341884**	**0.452358**	0.509590	**0.297789**	**2.396134**
	Distance	0.014576	0.012321	0.033020	0.719610	0.025659	0.049336
ABK	Dice	**2.0**	0.574338	1.867298	**2.0**	1.944250	**2.0**
	Jaccard	**1.0**	0.287169	0.933649	**1.0**	0.972125	**1.0**
	Beta	**0.583689**	0.007120	0.082563	0.086043	0.080972	0.069305
	Distance	0.054810	0.011901	0.010206	**0.005618**	0.006778	**0.007700**
ABFCM	Dice	**2.0**	0.740229	**1.923076**	**2.0**	1.944250	**2.0**
	Jaccard	**1.0**	**0.370114**	**0.961538**	**1.0**	0.972125	**1.0**
	Beta	0.233170	**0.341884**	0.290176	0.086043	0.080972	0.556407
	Distance	**0.006369**	0.012321	**0.006887**	**0.005618**	**0.006778**	0.016726

5.2.2. Performance based on Twenty Iterations

Table 4 explains the performance comparison in terms of dice coefficient, jaccard coefficient, beta index and distance index with twenty iterations and cluster sizes of three. In this Table 4, the performance of every algorithm in terms of dice coefficient and jaccard coefficient are similar using the dataset abalone. For abalone dataset, ABK algorithm shows better performance in terms of beta index when compare to other algorithms. When compared to other algorithms for distance index using the dataset abalone, AB-FCM shows better performance.

The performance comparison using the dataset zoo shows that the AB-FCM gives better performance in terms of dice coefficient compared to other algorithms. In terms of jaccard coefficient using the dataset zoo, the AB-FCM shows better performance contrast to other algorithms. Using the dataset zoo, the ABK algorithm performed well in terms of beta index compared to the other algorithms. Considering the dataset iris, the performance of ABFCM is better when compared to the other algorithms in terms of dice coefficient, jaccard coefficient and distance index. Using the dataset iris, the performance of ABK algorithm is better in terms of beta index.

Table 4. Performance comparison based on twenty iterations and three clusters.

Iteration=20, Cluster=3							
		Abalone	**Zoo**	**Iris**	**Wine**	**Liver**	**Thyroid**
K-Means	**Dice**	2.0	0.598870	1.869767	2.0	1.950920	2.0
	Jaccard	1.0	0.299435	0.934883	1.0	0.975460	1.0
	Beta	0.281168	0.169624	0.262697	0.385807	**1.345056**	0.413278
	Distance	0.017483	**0.007652**	0.020734	0.011928	3.889364	0.027379
BFO	Dice	2.0	0.653721	1.900621	2.0	1.948051	2.0
	Jaccard	1.0	0.326860	0.950310	1.0	0.974025	1.0
	Beta	0.059906	0.062109	0.059757	0.0612	0.061474	0.047561
	Distance	0.017410	0.028525	0.022202	0.015280	0.020017	0.006692
ABC	Dice	2.0	0.402297	1.870370	2.0	**1.951515**	2.0
	Jaccard	1.0	0.270118	0.935185	1.0	**0.975757**	1.0
	Beta	0.239303	0.321884	0.452358	0.509590	0.297789	2.396134
	Distance	0.014576	0.123210	0.033020	0.719610	0.025659	0.049336
ABK	Dice	**2.0**	0.577433	1.867298	**2.0**	1.944250	**2.0**
	Jaccard	**1.0**	0.287169	0.933649	**1.0**	0.972125	**1.0**
	Beta	**0.583689**	**0.712079**	**0.825639**	0.604318	0.809264	0.069305
	Distance	0.054810	0.011901	0.010206	0.015631	**0.006778**	0.007700
ABFCM	Dice	**2.0**	**0.740229**	**1.923076**	**2.0**	1.944250	**2.0**
	Jaccard	**1.0**	**0.370117**	**0.961538**	**1.0**	0.972125	**1.0**
	Beta	0.233170	0.341884	0.290176	**0.860431**	0.809726	**0.556407**
	Distance	**0.006369**	0.012321	**0.006887**	**0.005618**	**0.006778**	0.016726

Considering the wine dataset, the performance of every algorithm is similar in terms of dice coefficient and jaccard coefficient. In terms of Beta Index and Distance Index, the ABFCM algorithm shows better performance using the dataset wine. Using the dataset liver, the ABK and ABFCM algorithm shows better performance in terms of distance index compared to the other algorithms. When comparing the performance using the dataset thyroid in terms of dice coefficient and jaccard coefficient, all the algorithms used for comparison show similar performance. In terms of beta index using the thyroid dataset, the ABFCM algorithm shows better performance than the other algorithms.

This research has been performed to find the effective clustering method from the partitional and optimization-based clustering techniques. The research finding is that the hybrid algorithm (ABC with partitional clustering) provides better results in clustering compared with the partitional clustering as well as optimization-based clustering. This finding can lead to hybridization of various optimization algorithms with the partitional clustering.

6. Conclusions

In this paper, an extensive performance analysis for the comparison of different techniques such as BFO algorithm, ABC algorithm, ABFCM algorithm, ABK algorithm and K-means algorithm is performed. The performance of these techniques was calculated based on the distance metrics namely dice coefficient, jaccard coefficient, beta index and distance index with different iterations and different cluster size. The evaluation metrics are computed for six different datasets using the algorithms considered for comparison. From the experimental result, the performance of ABK and ABFCM algorithm outperformed in most of the cases compared with the existing algorithms. The future work can be in the direction of reducing the computational complexity that can be achieved with a best set on initial solutions.

References

[1] Ahmed M., Yamany S., Mohamed N., Farag A., and Moriarty T., "A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data," *IEEE Transactions on Medical Imaging*, vol. 21, no. 3, pp. 193-199, 2002.

[2] Bezdek J., *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press, 1981.

[3] Breunig M., Kriegel P., Kroger P., and Sander J., "Data Bubbles: Quality Preserving Performance Boosting for Hierarchical Clustering," *in Proceedings of ACM SIGMOD*, pp. 79-90, 2001.

[4] Chuang K. and Chen M., "Clustering Categorical Data by Utilizing the Correlated Force Ensemble," available at: http://arbor.ee.ntu.edu.tw/~doug/paper/sdm04.pdf , last visited 2008.

[5] Dubes R., "How Many Clusters Are Best?-An Experiment," *Pattern Recognition*, vol. 20, no. 6, pp. 645-663, 1987.

[6] Hertz J., Krogh A., and Palmer R., *Introduction to the Theory of Neural Computation. Reading*, Mass Addison-Wesley, 1991.

[7] Ilango M. and Mohan V., "PCFA: Mining of Projected Clusters in High Dimensional Data Using Modified FCM Algorithm," *the International Arab Journal of Information Technology*, vol. 11, no. 2, pp. 168-177, 2012.

[8] Jain A., Murty M., and Flynn P., "Data Clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, 1999.

[9] Karaboga D. and Ozturk C., "A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm," *Applied Soft Computing*, vol. 11, no. 1, pp. 652-657, 2011.

[10] Karmakar G. and Dooley L., "A Generic Fuzzy Rule Based Image Segmentation Algorithm," *Pattern Recognition Letters*, vol. 23, no. 10, pp. 1215-1227, 2002.

[11] Krishna K. and Narasimha M., "Genetic K-means Algorithm," *IEEE Transactions on Systems Man and Cybernetics B Cybernetics*, vol. 29, no. 3, pp. 433-439, 1999.

[12] Mcqueen J., "Some Methods for Classification and Analysis of Multivariate Observations," *in Proceedings of the 5th Berkeley Symposium on Math. Statistics and Probability*, pp. 281-297, 1967.

[13] Mualik U. and Bandyopadhyay S., "Genetic Algorithm Based Clustering Technique," *Pattern Recognition*, vol. 33, no. 9, pp. 1455-1465, 2002.

[14] Noordam J., Van Den W., Buydens L., "Geometrically Guided Fuzzy C-Means Clustering for Multivariate Image Segmentation," *in Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, pp. 462-465. 2000.

[15] Silahtaroglu G., "Clustering Categorical Data using Hierarchies," *World Academy of Science, Engineering and Technology*, vol. 56, no. 64, pp. 334-339, 2009.

[16] Sneath A. and Sokal R., *Numerical Taxonomy*, London, Freeman, 1973.

[17] Teodorovic D., Lucic P., Markovic G, and Dell' Orco M, "Bee Colony Optimization: Principles and Applications," *in Proceeding of the 8th Seminar on Neural Network Applications in Electrical Engineering*, Belgrade, pp. 151-156, 2006.

[18] Tolias Y. and Panas S., "Image Segmentation by a Fuzzy Clustering Algorithm Using Adaptive Spatially Constrained Membership Functions," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 359-369, 1998.

[19] Udupa J. and Samarasekera S., "Fuzzy Connectedness and Object Definition: Theory, Algorithm, and Applications in Image Segmentation," *Graphical Models and Image Processing*, vol. 58, no. 3, pp. 246-261, 1996.

[20] Wan M., Li L., Xiao J., Wang C., and Yang Y., "Data Clustering using Bacterial Foraging Optimization," *Journal of Intelligent Information System*, vol. 38, no. 2, pp. 321-341, 2012.

[21] Yang M., Hu Y., Lin K., and Lin C., "Segmentation Techniques for Tissue Differentiation in MRI of Ophthalmology using Fuzzy Clustering Algorithms," *Magnetic Resonance Imaging*, vol. 20, no. 2, pp. 173-179, 2002.

Krishnamoorthi Murugasamy obtained his BE from Bharathiyar University, India in 2002. Then he obtained his ME CSE from Annamalai University, Chidambaram, India in 2004 and currently pursing PhD in Computer Science majoring in Data mining from Anna University of Technology. He is currently working as Assistant Professor-Senior Grade in the Department of Computer Science and Engineering, Bannari Amman Institute of Technology, Erode. His area of interest includes database management system, data mining, computer networks and network security. His current research interests are data clustering and optimization techniques.

Natarajan Mathaiyan Professor and Chief Executive, Bannari Amman Institute of Technology, India. He received his BE, the MSc and PhD degree from PSG College of Technology, India in 1968, 1970 and 1994 respectively. He has 42 years of experience in Academic-Teaching, Research and Administration. He is the member in the Academic Council of the Anna University, India. He is also, a member in the Board of Studies, Anna University Chennai. He is a member in ISTE, IEEE, IEI, ACM, CSI, CII etc. He had published 45 papers in national and international journals and 85 papers in the national and international conferences. He authored and published 10 Books.