Fast Efficient Clustering Algorithm for Balanced Data

Adel A. Sewisy
Computer Science Department
Faculty of Computer and Information, Assiut University
Assiut, Egypt

M. H. Marghny
Computer Science Department
Faculty of Computer and Information, Assiut University
Assiut, Egypt

Rasha M. Abd ElAziz
Computer Science Department
Faculty of Science, Assiut University
Assiut, Egypt

Ahmed I. Taloba
Computer Science Department
Faculty of Computer and Information, Assiut University
Assiut, Egypt

***Abstract*—The Cluster analysis is a major technique for statistical analysis, machine learning, pattern recognition, data mining, image analysis and bioinformatics. K-means algorithm is one of the most important clustering algorithms. However, the k-means algorithm needs a large amount of computational time for handling large data sets. In this paper, we developed more efficient clustering algorithm to overcome this deficiency named Fast Balanced k-means (FBK-means). This algorithm is not only yields the best clustering results as in the k-means algorithm but also requires less computational time. The algorithm is working well in the case of balanced data.**

***Keywords*—Clustering; K-means algorithm; Bee algorithm; GA algorithm; FBK-means algorithm**

I. INTRODUCTION

The problem of clustering is perhaps one of the most widely studied in the data mining and machine learning communities. This problem has been studied by researchers from several disciplines over five decades. Applications of clustering include a wide variety of problem domains such as text, multimedia, social networks, and biological data. Furthermore, the problem may be encountered in a number of different scenarios such as streaming or uncertain data. Clustering is a rather diverse topic, and the underlying algorithms depend greatly on the data domain and problem scenario [1-6].

The goal of clustering is to group a set of objects into clusters so that the objects in the same cluster are highly similar but remarkably dissimilar with objects in other clusters. To tackle this problem, various types of clustering algorithms have been developed in the literature. Among them, the k-means clustering algorithm [7] is one of the most efficient clustering algorithms for large-scale spherical data sets. It has extensive applications in such domains as financial fraud, medical diagnosis, image processing, information retrieval, and bioinformatics [8]. Several clustering algorithms have been developed yet, most of them could not fulfill the requirements of clustering problem which are [9]:

a) High dimensionality: Natural clusters usually do not exist in the full dimensional space, but in the subspace formed by a set of correlated dimensions. Locating clusters in subspaces can be challenging.

b) Scalability: Real world data sets may contain hundreds of thousands of instances. Many clustering algorithms work fine on small data sets, but fail to handle large data sets efficiently.

c) Accuracy: A good clustering solution should have high intra-cluster similarity and low Inter-cluster similarity.

The k-means algorithm and its approaches are known to be fast algorithms for solving such problems. However, they are sensitive to the choice of starting points and can only be applied to small datasets [10].

One common way of avoiding this problem is to use the multi restarting k-means algorithm. However, as the size of the dataset and the number of clusters increase, more and more starting points are needed to get a near global solution to the clustering problem. Consequently the multi restarting k-means algorithm becomes very time consuming and inefficient for solving clustering problems, even in moderately large datasets [11].

In this paper, a new clustering algorithm is proposed for clustering large data sets called FBK-means. The algorithm minimizes an objective function to determine new cluster centers. Compared with the K-means algorithm and other existing modifications, the FBK-means algorithm can obtain a slightly better result but with a lower computational time. The algorithm is working well in the case of balanced data.

The rest of this paper is organized as follows. Section 2 reviews the k-means algorithm and some existing modifications. Section 3 presents a more efficient FBK-means algorithm. Section 4 analyzes the results of the proposed algorithm. Finally, Section 5 concludes the paper with some remarks.

II. BACKGROUND

In this section, we give a brief description of the k-means and some existing modifications.

A. K-means algorithm

K-means algorithm is one of the most popular clustering algorithms and is widely used in a variety of fields. In k-means algorithm, a cluster is represented by the mean value of data points within a cluster and the clustering is done by minimizing the sum of distances between data points and the corresponding cluster centers. Typically, the squared-error (SE) criterion is used, defined as:

$$SE = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \left\| x_j - m_i \right\|. \quad (1)$$

Where SE, is the sum of square-error for all objects in the dataset, k number of clusters, n_i number of objects in each cluster, x_i is the point in space representing a given object, and m_i is the mean of cluster c_i.

The validity of all clusters is defined as [12]:

$$\text{Validity} = \frac{\text{Inter\_Cluster\_Dist}}{\text{Intra\_Cluster\_Dist}}. \quad (2)$$

Where the intra-cluster distance is defined as:

$$Intra\_Cluster\_Dist = \frac{1}{N} \sum_{i=1}^{k} \sum_{x_j \in S_i} \left| x_j - \mu_i \right|^2 \quad (3)$$

Where N is the total number of data points, S_i, i = 1,2,.., k, are the k clusters and μ_i is the centroid or mean point of all the points $x_i \in S_i$. Another measure of cluster performance is the inter-cluster distance, i.e., the distance between clusters. This is calculated by taking the minimum of the distances between each pair of cluster centroids as follows:

$$Inter\_Cluster\_Dist = \min_{j=i+1,\ldots,k} \left(\left| \mu_i - \mu_j \right|^2 \right), \quad i = 1, 2, \ldots, k-1 \quad (4)$$

We take the minimum of the distance between clusters because it is the upper limit of cluster performance and is expected to be maximized. The ratio of intra-cluster distance to inter-cluster distance can serve as an evaluation function for cluster performance. Since we want to maximize the inter-cluster distance and minimize the intra-cluster distance, we want the validity value to be maximized.

The steps of the k-means algorithm are as follows:

Step 1: Choose a seed solution consisting of k centers (not necessarily belonging to A).

Step 2: Allocate data points a ∈ A to its closest center and obtain k-partition of A.

Step 3: Re-compute centers for this new partition and go to Step 2 until no more data points change their clusters.

This algorithm is very sensitive to the choice of a starting point. It converges to a local solution, which can significantly differ from the global solution in many large data sets. The running time of k-means algorithm grows with the increase of the size and dimensionality of the data set. Hence, clustering of large dataset consumes a great time large error.

Many of the methods discussed these problems, but each method has been focusing on a specific problem, the most important of these methods are genetic clustering algorithm (GA) and Bee-clustering algorithm (Bee).

A. Genetic Clustering Algorithm (GA)

Genetic algorithm [13] is a very popular evolutionary algorithm, formatted by simulating the principle of survival of the fittest in natural environment. It mainly include genes coding, fitness calculations, creating the initial population, determine the evolutionary operation etc, which mainly include selection, crossover and mutation.

The steps of the genetic clustering algorithm are as follows:

Step 1: Set the parameters: population size M, the maximum number of iteration T, the number of clusters K, etc.

Step 2: Generate m chromosomes randomly; a chromosome represents a set of initial cluster centers, to form the initial population.

Step 3: According to the initial cluster centers showed by every chromosome, carry out k-means clustering, each chromosome corresponds to once k-means clustering, then calculate chromosome fitness in line with clustering result, and implement the optimal preservation strategy.

Step 4: For the group, to carry out selection, crossover and mutation operator to produce a new generation of group.

Step 5: To determine whether the conditions meet the genetic termination conditions, if meet then withdrawal genetic operation and tum 6, otherwise tum 3.

Step 6: Calculate fitness of the new generation of group; compare the fitness of the best individual in current group with the best individual's fitness so far to find the individual with the highest fitness.

Step 7: Carry out k-means clustering according to the initial cluster center represented by the chromosome with the highest fitness, and then output clustering result.

A. Bee Clustering Algorithm (Bee)

The Bee Algorithm [14] is an optimization algorithm inspired by the natural behaviour of honey bees to find an optimal solution.

The steps of the Bee clustering algorithm are as follows:

Step 1: Generate initial population of solutions randomly (n).

Step 2: Evaluate each solution using the fitness function.

Repeat the following steps until stopping criterion is met:

Step 3: Select the best solutions (m) for neighborhood search

Step 4: Assign more bees to the ones with highest fitness's (e) out of best solutions (m).

Step 5: Select the fittest bee from each patch.

Step 6: Assign the remaining bees for random search and evaluate their fitness.

III. Fast Balanced K-Means Algorithm

In the k-means algorithm, cluster results depends on the random initial centers. Many developed algorithms try to solve this problem but these algorithms rely idea solution it assumes a population of solutions then select the best and try to find another solutions from them. These steps carried out more than once, until get the best solution and this consumes more time with the increase of the size and dimensionality of the datasets.

To solve these problems we proposed a new fast efficient clustering algorithm for clustering large datasets called FBK-means. This algorithm not only obtain a better result but also with a lower computational time.

The idea of this algorithm, first we generate K random centers then assign each point to its closest center to obtain K clusters, second we compute the validity for each cluster $V_i = integer\left(\frac{D_i - AVG}{Rate}\right)$ where $i = 1,2,\dots,k$, D_i is the sum of distance for cluster i, AVG the average distance for all clusters and the Rate=ER*AVG where ER is error rate. When V_i is positive this means there are overlapping between two or more clusters, if V_i is negative this means there are two or more centers in one cluster and if V_i is zero this means the cluster is better.

Depends on V_i, for each iteration we try to improve the validity of all clusters as in equation (2). By moving the center of cluster that has smallest negative V_i to the cluster that has large positive V_i as in Fig. 1, until all V for all clusters equal zero. Finally apply k-means algorithm to the final centers obtained.

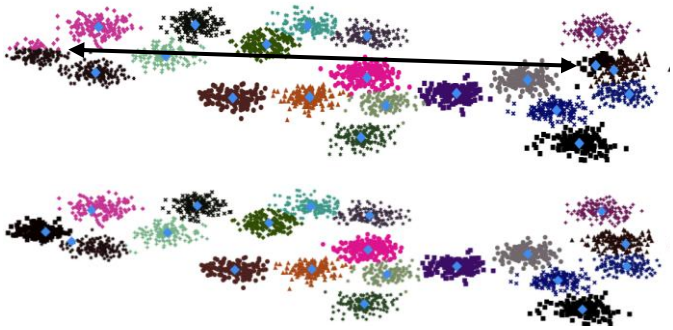

Fig. 1. Moving center from cluster to another for A1 dataset.

The FBK-means algorithm can be summarized as follows:

Step 1: Generate K random centers.

Step 2: For each point a ∈ A Assign its closest center and obtain k clusters of A.

Step 3: Compute the validity of all clusters as in equation (2) and the sum of distance for each cluster D_i, i =1,2,…, k.

Step 4: Compute the average distance for all clusters

$$Avg = \frac{\sum D_i}{k}, i = 1,2,\dots,k$$

Step 5: for each cluster compute:

a- Rate = ER * Avg

b- Validity for each cluster

$$V_i = integer\left(\frac{D_i - AVG}{Rate}\right), i = 1,2,\dots,k$$

Step 6: for each cluster i do one of this:

a) If D_i = 0 (empty cluster) move the center of this cluster to the cluster that has large positive validity.

b) Else if V_i < 0 then moving the center of cluster that has smallest negative validity to the cluster that has large positive validity.

c) Else compute new center to this cluster by computing the average of all of the points of this cluster

Step 7: Repeat Steps 2 to 7 until all V elements equal zero, or the difference between E for each iteration less than the threshold.

Step 8: Apply k-means algorithm to the final centers and compute SE (1) of final results.

Where A is the data, K number of clusters, ER is the error rate, SE is the squared-error, E is the validity of all clusters, D_i, i =1,2,…, k is the sum of distance for each cluster and V_i , i =1,2,…, k is validity of cluster i.

IV. Experimental Results

In order to evaluate the performance of the proposed algorithm, we compare the results with the results of k-means, GA and Bee algorithms on synthetic datasets [15], which are shown in Table I. The implementations have been done in visual studio 2010 on windows7 running on a PC with an Intel core2 duo processor (2.13GHz) with 4GB RAM.

The GA parameters that have been used in the experimental: the population size = 10, selection is roulette, crossover is single point crossover, the probability of crossover = 0.8 and the probability of mutation = 0.001. The Bee parameters that have been used: number of scout bees (n) = 10, number of sites selected for neighborhood searching (m) =5 and number of top-rated (elite) sites among m selected sites (e) =2. For the FBK-means algorithm: the error rate (ER) = 0.2.

TABLE I. Summary of The Datasets

Datasets	Ins. No.	Cluster No.
A1	3000	20
A2	5250	35
A3	7500	50
S1	5000	15
S2	5000	15
S3	5000	15
S4	5000	15
Birch1	100000	100
Birch1	100000	100
Birch1	100000	100

TABLE II. THE AVERAGE SQUARE ERROR AFTER 5 ITERATIONS

Datasets	K-means	GA	Bee	FBK-means
A1	6672474	6314849	6210784	5376830
A2	11808602	11455847	11638661	9251513
A3	16277424	19553389	17054519	13086989
S1	273965201	241313731	247915460	169390458
S2	268798832	270339590	247915460	207120231
S3	274632946	280538768	274874787	241045713
S4	259292277	249340132	240559431	237230705
Birch1	3170853720	3227020055	3115915012	2754436631
Birch2	339313573	306477071	306942942	189361571
Birch3	1830504972	1919705867	1813960089	1621350273

TABLE III. THE AVERAGE TIME IN SECONDS AFTER 5 ITERATIONS

Datasets	K-means	GA	Bee	FBK-means
A1	5	22	67	3
A2	10	69	181	12
A3	64	179	348	19
S1	8	57	64	10
S2	11	68	64	10
S3	7	60	72	16
S4	19	28	67	20
Birch1	418	3094	6845	635
Birch2	297	3070	7005	580
Birch3	778	3599	5980	575

Table II shows the mean square errors after 5 iterations for the k-means algorithm, GA algorithm, Bee algorithm and FBK-means algorithm, it is obviously that the squared error obtained by FBK-means algorithm is better with a lower computational time see Table III.

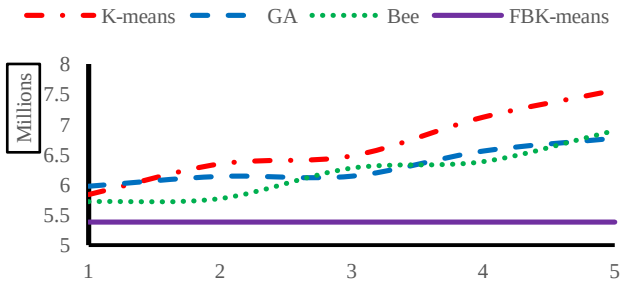

Fig. 2. Square error for A1 after 5 iterations

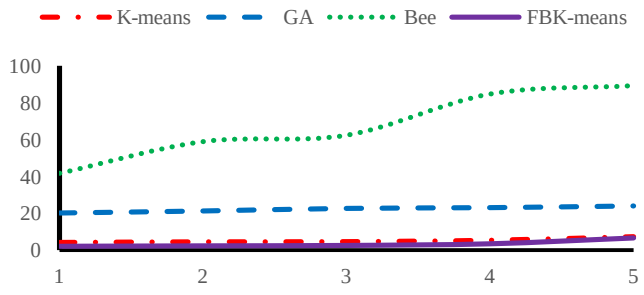

Fig. 3. Time in seconds for A1 after 5 iterations

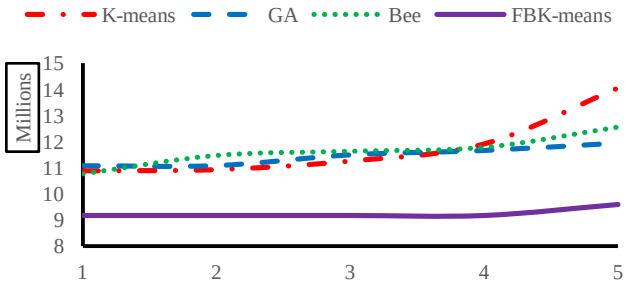

Fig. 4. Square error for A2 after 5 iterations

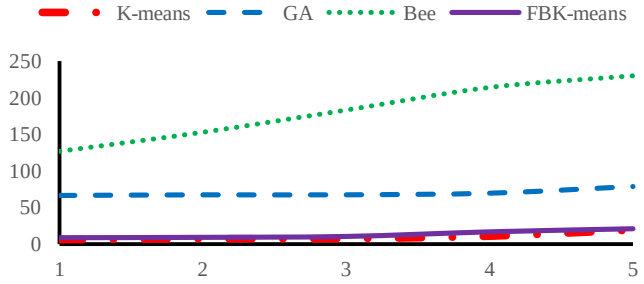

Fig. 5. Time in seconds for A2 after 5 iterations

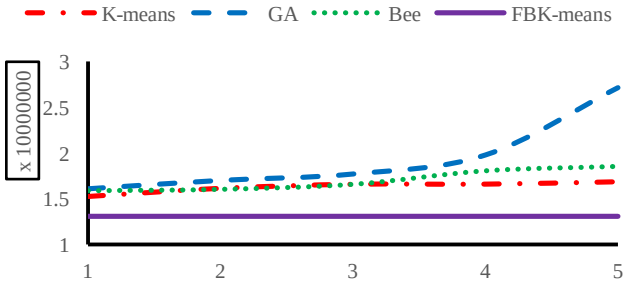

Fig. 6. Square error for A3 after 5 iterations

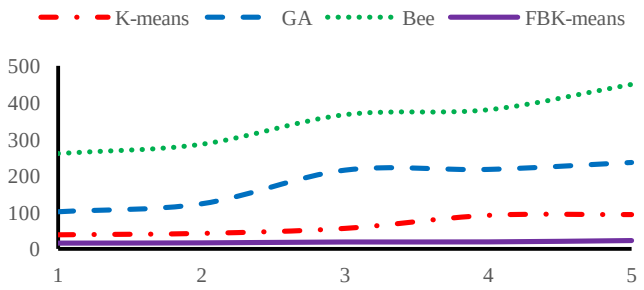

Fig. 7. Time in seconds for A3 after 5 iterations

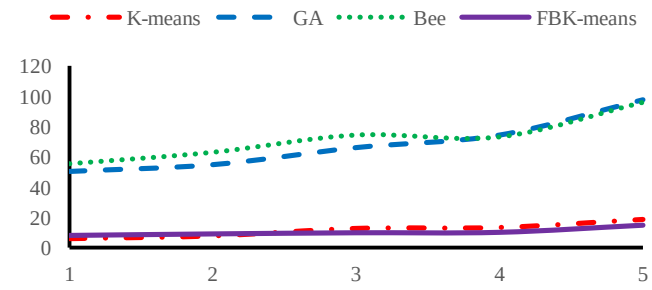

Fig. 11. Time in seconds for S2 after 5 iterations

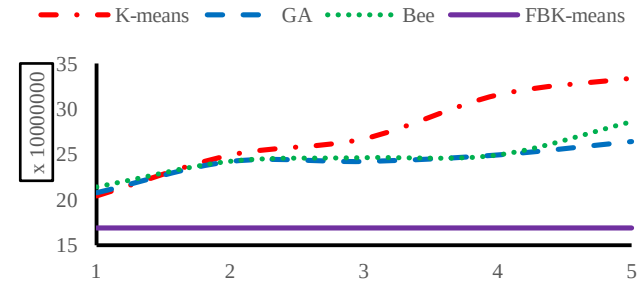

Fig. 8. Square error for S1 after 5 iterations

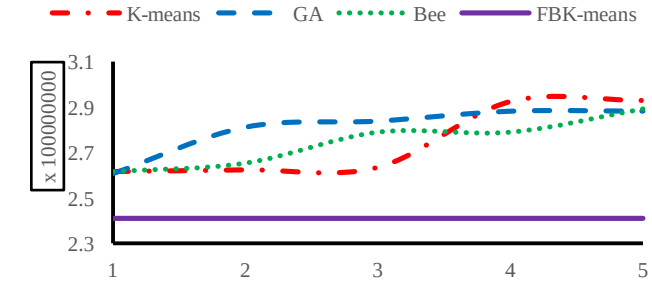

Fig. 12. Square error for S3 after 5 iterations

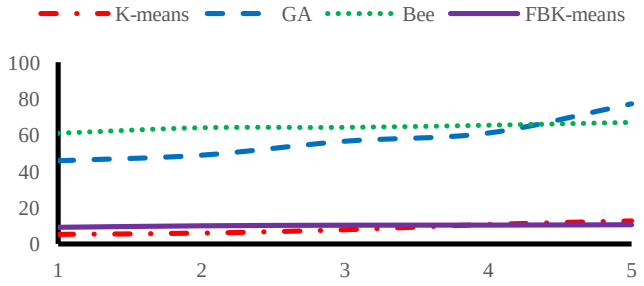

Fig. 9. Time in seconds for S1 after 5 iterations

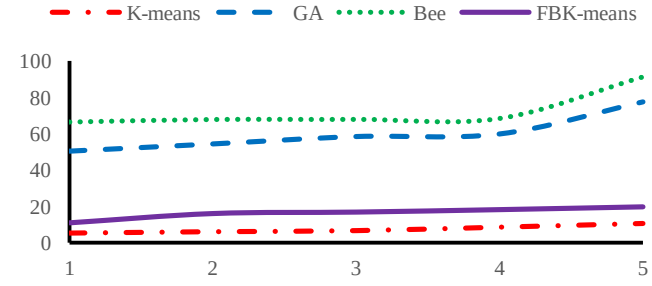

Fig. 13. Time in seconds for S3 after 5 iterations

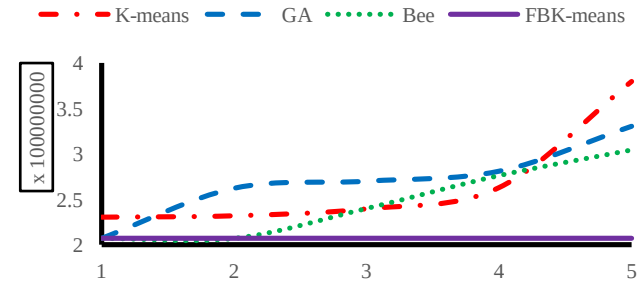

Fig. 10. Square error for S2 after 5 iterations

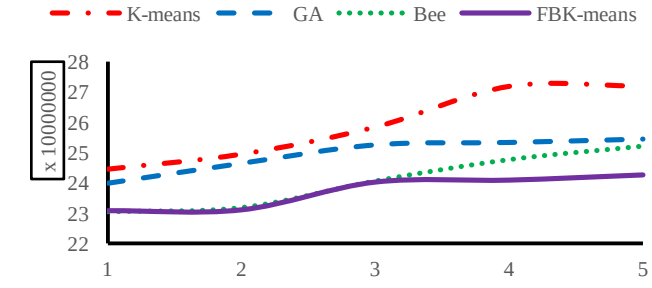

Fig. 14. Square error for S4 after 5 iterations

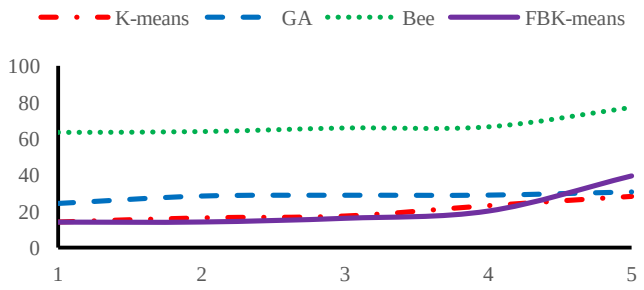

Fig. 15. Time in seconds for S4 after 5 iterations

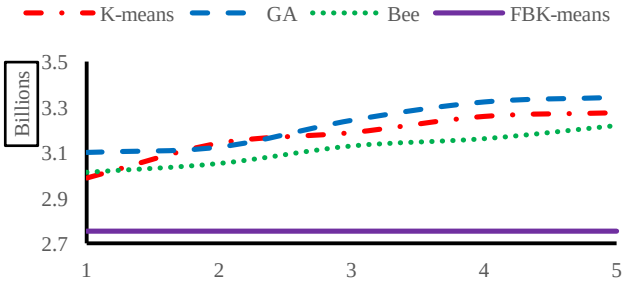

Fig. 16. Square error for Birch1 after 5 iterations

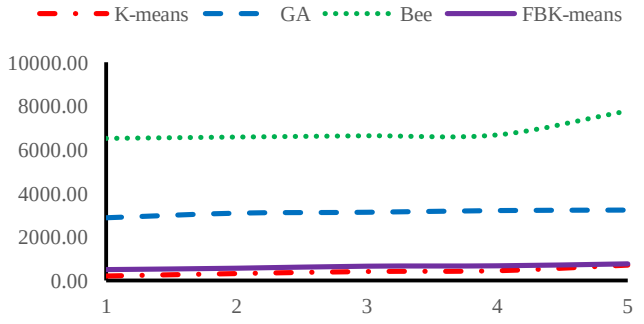

Fig. 17. Time in seconds for Birch1 after 5 iterations

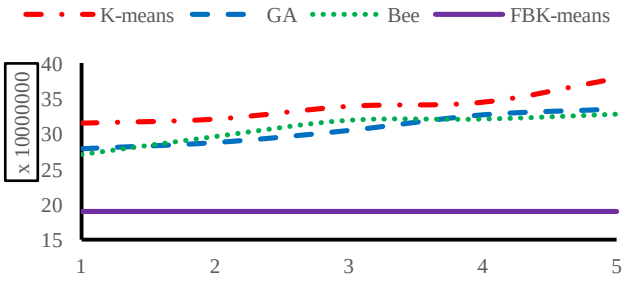

Fig. 18. Square error for Birch2 after 5 iterations

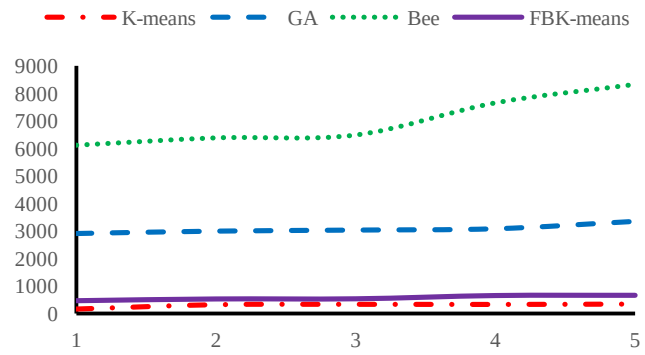

Fig. 19. Time in seconds for Birch2 after 5 iterations

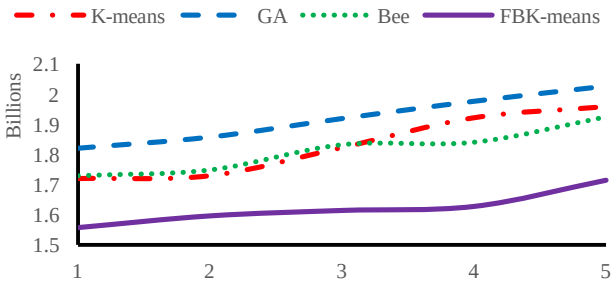

Fig. 20. Square error for Birch3 after 5 iterations

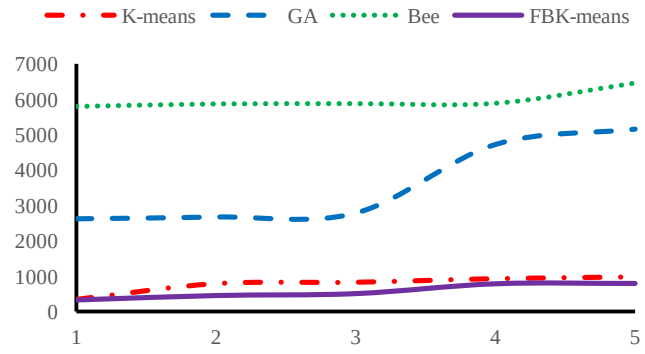

Fig. 21. Time in seconds for Birch3 after 5 iterations.

Fig. 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20 shows the mean square errors for ten datasets after 5 iterations, respectively. Fig. 3, 5, 7, 9, 11, 13, 15, 17, 19 and 21 shows the execution time for ten datasets after 5 iterations, respectively.

The results show that the FBK-means algorithm outperforms the k-means, GA and Bee algorithms in terms of computing time and square error calculations. When the number of dimensions or clusters k increases, the efficiency of the proposed algorithm becomes more remarkable than the k-means, GA and Bee algorithms.

V. CONCLUSION

To improve the efficiency of the k-means clustering algorithm, a new clustering algorithm is proposed for clustering large datasets called FBK-means. This algorithm minimizes an objective function to determine new cluster centers. Compared with the k-means, GA and Bee algorithms, FBK-means algorithm requires less computing time and fewer distance calculations while retaining the same clustering results. The performance of the proposed algorithm is more remarkable as the number of dimensions or clusters of a dataset increases.

REFERENCES

[1] C.C. Aggarwal and C.K Reddy, "Data clustering: algorithms and applications", Chapman and Hall/CRC Press, 2013.

[2] M.H.Margahny and A.A.Mitwaly, "Fast Algorithm for Mining Association Rules", AIML 05 Conference, Cairo, Egypt, 2005

[3] Marghny, M. H., and A. A. Shakour, "Fast, Simple and Memory Efficient Algorithm for Mining Association Rules", International Review on Computers & Software, 2007.

[4] Margahny, M. H., and A. A. Shakour, "Scalable Algorithm for Mining Association Rules", ICCST, 2006.

[5] M. H. Marghny, Rasha M. Abd El-Aziz and Ahmed I. Taloba, "An Effective Evolutionary Clustering Algorithm: Hepatitis C Case Study", Computer Science Department, Egypt, International Journal of Computer Applications, vol. 34, No.6, pp. 0975-8887, 2011.

[6] M. H. Marghny and Ahmed I. Taloba, "Outlier Detection using Improved Genetic K-means", International Journal of Computer Applications, vol. 28, No.11, pp. 33-36, 2011.

[7] J. MacQueen, "Some methods for classification and analysis of multivariate observations", Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297, 1967.

[8] L. Ba, J. Liang, C. Sui and D. Dang, "Fast global k-means clustering based on local geometrical information", Information Sciences, vol. 245, pp.168-180, 2013.

[9] N. M. Abdel-Hamid, M.B. Abdel-Halim and M. W. Fakhr, "Bees algorithm-based document clustering", ICIT The 6th International Conference on Information Technology , 2013.

[10] A. Bagirov, J. Ugon and D. Webb, "Fast modified global k-means algorithm for incremental cluster construction", Pattern Recognition, vol. 44, pp.866–876, 2011.

[11] A. Bagirov, J. Ugon and D. Webb, "A new modified global k-means algorithm for clustering large data sets", ASMDA The XIII International Conference , pp.1-5, 2009.

[12] L. An, H. Xie, M. Chin, Z. Obradovic, D. Smith and V. Megalooikonomou, "Analysis of multiplex gene expression maps obtained by voxelation", Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp.23-28, 2008.

[13] W. Min and Y. Siqing, "Improved k-means clustering based on genetic algorithm", Computer Application and System Modeling (ICCASM), vol. 6, pp.636–639, 2010.

[14] D. T. Pham, S. Otri, A. Afify, M. Mahmuddin, and H. Al-Jabbouli, "Data clustering using the bees algorithm", 40th CIRP International Manufacturing Systems Seminar, 2007.

[15] "http://cs.joensuu.fi/sipu/datasets/"