Int J Syst Assur Eng Manag
DOI 10.1007/s13198-017-0665-x

ORIGINAL ARTICLE

Dynamic frequency based parallel k-bat algorithm for massive data clustering (DFBPKBA)

Ashish Kumar Tripathi[1] · **Kapil Sharma**[1] · **Manju Bala**[2]

Received: 22 March 2017 / Revised: 12 June 2017

Abstract In the past one decade there has been significant increase in the growth of digital data. Therefore, good data mining techniques are important for the better decision making. Clustering is one of the key element in the field of data mining. K-means is a very popular algorithm present in the literature which is widely used for the clustering purpose. However k-means algorithm suffers from the problem of stucking into local optimum solution because of it's dependency on the random initialization of initial cluster center. In this paper a novel variant of Bat algorithm based on dynamic frequency is introduced. Further the proposed variant is hybridized with K-means to present a new approach for clustering in distributed environment. Since evolutionary computation is very computation intensive, traditional sequential algorithms are not able to provide satisfactory results within the reasonable amount of time for the large scale data problems. To mitigate this problem the proposed variant is parallelized using the MapReduce model in the Hadoop framework. The experimental results show that the proposed algorithm has outperformed K-means, PSO and Bat algorithm on eighty percent of the benchmark datasets in terms of intra-cluster distance. Further DBPKBA has also achieved significant speedup for dealing with massive datasets with increase in the number of nodes.

✉ Kapil Sharma
kapil@ieee.org

Ashish Kumar Tripathi
mail2ashish07@gmail.com

[1] Delhi Technological University, Delhi, India

[2] IP College of Women, Delhi, India

Keywords Bat algorithm · Hadoop · MapReduce · Large data sets · DFBPKBA

1 Introduction

With the progress of technology there has been a significant increase in the growth of the digital data. Data mining techniques have automated the task of deriving the meaningful conclusions from large datasets in a short time frame. Clustering is one of the popular data mining technique that groups data items in multiple clusters with maximum intra-cluster similarity and minimum inter-cluster similarity (Lin et al. 2004) . Data clustering is widely used in the many domains like image segmentation, data mining, bio-medical and information retrieval (Fayyad et al. 2002). The traditional clustering algorithms are failing with the increasing size of the data. Therefore, improvement in the traditional sequential computational model is required to cluster massive data sets in reasonable amount of time.

K-means is a popular and simple clustering algorithm used in the various clustering applications (Xu and Wunsch 2005). However, k-means algorithm suffers from the following problems:

- The quality of clusters in k-means algorithm is highly sensitive towards the initial cluster center values.
- There is no information sharing because of which this algorithm easily falls into the local optimum solution.

To overcome these problems, many researchers now a days are using nature inspired algorithms with k-means for the clustering purpose. This hybridization of k-means have shown ability to find the global best solution (Hatamlou

Published online: 06 September 2017

et al. 2012) and avoided the problem of trapping into local optima. Although these nature inspired algorithms have overcome the shortcoming of k-means algorithm but still not preferred for clustering massive datasets due to their computation intensiveness. Parallelization of the these algorithms is one of the promising solution to handle the large scale data. So, various distributed computing platforms such as cuda, gpu and Hadoop (Gong et al. 2015) are used to reduce the computational cost.

Gong et al. (2015) surveyed and compared different distributed models used in parallelization of nature inspired algorithms for handling large datasets. Among all studied models, hadoop map reduce architecture is one of the popular platform because of its simplicity scalability and robustness. It was concluded that parallelization with Master-Slave model of the Hadoop with MapReduce architecture has the highest fault tolerance and implementation simplicity while CUDA based model is difficult and less fault tolerant but have better execution time. Hadoop is an open source software developed by apache foundation which works on the MapReduce programming architecture that provides parallel computing environment on Hadoop distributed file system termed as HDFS (Frontpage 2016). MapReduce architecture works by dividing the data set into small blocks and send it to the map function running on various nodes. The map function calculate the distance of data point and decides that to which cluster the particular data point will belong. Reduce function then combines the Map function results from the various machines to calculate the final cluster center location.

This paper presents a novel algorithm for the efficient and fast clustering of large scale data using bat algorithm on Map-reduce architecture, namely dynamic frequency based parallel kbat algorithm (DFBPKBA).The proposed DFBPKBA algorithm is incorporated with the following capabilities.

- k means is used to initialize the cluster center to make convergence faster.
- Frequency tuning of the original Bat algorithm is modified to enhance the poor capability of exploitation and exploration of the of the Bat algorithm.
- The proposed Hybrid algorithm is parallelized using the MapReduce architecture to accelerate the speed of computation and reduce the computation time and find better results for the massive data sets.

The proposed algorithm leverages the strength of k-means by proposing a new variant of bat algorithm with the MapReduce programming model. This improved the clustering results in terms of and speed and quality of clusters generated. DFBPKBA has comparatively better exploration and exploitation capability than standard bat algorithm by modifying the frequency parameter. Rest of the paper is described as follows: Sect. 2 describes the work done in the field parallel clustering, Sect. 3 gives the background study of the clustering and k-means algorithm, Sect. 4 describes the proposed (DFBPKBA). The experimental settings and the results of the proposed approach are discussed in Sect. 5.

2 Related work

In this section various meta-heuristic algorithms used for the clustering purpose are discussed along with their parallelization on the MapReduce platform. Hadoop and MapReduce model have been widely used by a number of researchers for parallelization and solving computational intensive tasks (Khezr and Navimipour 2015; Wang et al. 2012). Paper (Khezr and Navimipour 2015) have given a survey of the MapReduce based parallel optimization algorithm. It was concluded in the paper that the MapReduce platform is being widely used now a days for the fast processing of the complex problems.

Paper (Wang et al. 2012) proposed k-PSO algorithm which combined Particle Swarm Optimization (PSO) with K-means using MapReduce. The proposed k-PSO made k-means parallel with MapReduce architecture for enhancing the possessing of massive data and improved global search with PSO. XY pang et al. proposed parallel genetic algorithm using the MapReduce architecture (Verma et al. 2009). In the proposed algorithm, population initialization and fitness calculation was done inside the Mapper. Each segment of chromosome was assumed to be the center of the cluster. In the second phase the reducer forms a new chromosome by joining the results received from the mapper. Process is repeated until all the centers of the chromosomes have not achieved the maximum inter cluster distance. Final chromosome contains the location of the optimal clusters.

Bhavani et al. (2011) proposed hybrid of k-means, differential evolution (DE) and ant colony optimization(ACO) using MapReduce architecture to identify species from their genome sequence. Feature descriptors for a genome sequence are identified using MapReduce architecture. Each feature descriptor is a three letter keyword, generated using A, T, C, G nucleotide bases. Genome sequences of related species are clustered by considering the feature descriptor count. It has been concluded that the inherent parallelism in the MapReduce model enhanced the execution time.

Aljarah and Ludwig (2013) used the MapReduce architecture for parallelization of particle swarm optimization and perform intrusion detection. Parallel architecture of the particle swarm optimization was able to handle and analyze high traffic data easily.

Nguyen et al. proposed a multi-objective firefly algorithm to solve the localization issues in WSN_s. The proposed approach improved the convergence rate and accuracy of the localization issues in the wireless sensor networks (Nguyen et al. 2016). Jagdish Chand Bansal et al. proposed spider monkey optimization algorithm for numerical optimization which mimics the foraging behavior of spider monkeys. The proposed algorithm was found to be competitive with PSO, DE, ABC and CMA-ES in terms of reliability, accuracy and efficiency (Bansal et al. 2014).

Tsai et al. proposed a variant of the original bat algorithm which was able to make a proper balance between exploration and exploitation. The proposed approach was tested on three bench mark functions with ten, thirty, fifty and hundred dimensions and it was concluded that new variant is able to enhance forty seven percent accuracy of the original bat algorithm (Tsai et al. 2016). The accuracy of the Evolved Bat algorithm was also improved in paper (Tsai et al. 2015; Cai and Tsai 2016). Kavita Sharma et al. developed a Fitness based particle swarm optimization to improve the slow convergence rate and stagnation in the local optima of the original PSO algorithm by incorporating a new position updating phase. The new phase was inspired from the onlooker phase of the Artificial Bee Colony algorithm. The proposed algorithm has outperformed original particle swarm optimization and artificial bee colony optimization in terms of mean fitness and convergence rate (Sharma et al. 2015).

Shimpi Singh Jadon et al. in their paper (Jadon et al. 2014) improved the problem of poor exploitation and slow convergence rate of Artificial Bee Colony Optimization. The proposed algorithm was compared with original ABC along with Best-So-Far ABC, Gbest guided, ABC, and Modified ABC on twenty four benchmark datasets. The proposed algorithm reduced up to half number of functions evaluations as compared to the original algorithm.

Jansen et al. (2009) investigated clustering of 150,000 microblog posts containing comments and sentiments regarding different brands and showed that microbiology can be a tool for marketing decisions. Ma et al. (2016) proposed a model based on neural network for identifying rumors in the large data sets like microblog.

B. Wu et al proposed MapReduced based ant colony optimization for solving combinatorial optimization problems (Wu et al. 2012) whereas paper (Xu et al. 2014; Lin et al. 2013) used MapReduce architecture for the parallelization of cuckoo search and concluded that MapReduce architecture is efficient for solving computation intensive problems. Paper (Moertini and Venica 2016) presented a parallel k-means algorithm for discovering knowledge from big data. The author showed that parallel k-means algorithm using MapReduce architecture can perform mining of big data in reasonable time.

In del Río et al. (2014), authors analyzed the techniques used for imbalanced big data processing using Random forest classifier.The problem of oversampling, under sampling and cost sensitive learning was adopted using MapReduce to handle the large data sets and to correctly identify the unrepresented class. It has been observed that the performance increases with increase in the number of mappers.

Meena et al. (2012) developed an enhanced parallel Ant Colony Optimization algorithm for text feature selection problem. However, the increased time complexity of the proposed algorithm was reduced by parallelizing with MapReduce architecture.

You et al. (2014) in their work proposed a parallel support vector machine (SVM) model for the prediction of large scale protein-protein interactions (PPI). Since the process of finding protein-protein interaction is computation intensive and complex, author employed the MapReduce based parallel architecture for training the SVM.This work achieved significant improvement in the training time and a good level of accuracy in predicting the PPI.

3 Background

3.1 Bat algorithm

The basic inspiration behind the bat algorithm is the echolocation behavior of the Micro-bats (Yang and He 2013). Microbats have ability to find their prey even in the darkness by using their echolocation behavior. Microbats emit sound pulses of different properties as per their hunting strategies and waits for emitted echo to bounced back from the surrounding objects (Yang 2010; Yang and He 2013). In bat algorithm, every bat at position X_i flies randomly in the search space to find the prey. The velocity V_i of bat is determined by fixed frequency f_i and loudness A_0. The loudness is assumed to vary from a large positive A_0 to a minimum constant value A_min. However, the wavelength and rate of pulse emission is adjusted automatically depending upon the distance from the prey. Bat algorithm makes a balance between exploration and exploitation through two parameters, i.e. frequency and emission rate. The frequency of ith bat is determined by Eq. (1)

$$freq_i = freq_{min} + (freq_{max} - freq_{min}) \times rand \quad (1)$$

$$vel_i^t = vel_i^{t-1} + (pos_i^{(t-1)} - pos^*) \times freq_i \quad (2)$$

$$pos_i^t = pos_i^{(t-1)} + vel_i^t \quad (3)$$

Here, x^* is the current global best solution among all the bats. The new equation of local search around the best is as follows:

$$pos_{new} = pos^* + \epsilon * 0.001 \tag{4}$$

Here, ϵ is a random number in the range $[-1, 1]$. Also, the average of loudness of bats has been skipped and instead the factor 0.001 has been used to multiply to the best solution. This has been done because searching for best solution should be done as close to the best value as possible. This is the base of exploitation. The equations for update of loudness and pulse rate are:

$$L_i^{t+1} = \alpha * L_i^t \tag{5}$$

$$rate_i^{t+1} = rate_0(1 - e^{-\gamma * t}) \tag{6}$$

In these equations, α is taken as 0.9 and λ is taken as 0.001 so that better exploitation can be done. Compared with the existing meta-heuristics, the bat algorithm has the advantage of dynamic control of exploitation and exploration by performing a local search around the best solution. When a random number generated becomes greater than the pulse rate value, the algorithm switches to exploitation around the best solution. But this structure of exploration and exploitation is not balanced as it should be. At the beginning of the algorithm, first exploitation is done and then exploration but it should be the other way round.

3.2 Clustering approach

Let $O = (o_1, o_2, o_3, \ldots, o_n)$ be a set of n data points where each data point is described by d features. These data points can be represented by a matrix of $O_{n\times d}$ that has n row vectors, where each row vector of d dimensions represents a particular data point. So, the ith row vector i.e $O_i = (o_i^1, o_i^2, o_i^3, \ldots, o_i^d)$ represents the ith data point of the data set, and each element o_i^j in O_i is scalar denoting the jth feature of the corresponding data point. The goal of a clustering algorithm is to find a set of cluster centers, $C = \{C_1, C_2, \ldots, C_k\}$ of k groups such that the data points belonging to the same cluster have maximum similarity, while the data points belonging to the different group have minimum similarity. The resulting group of data points is called clusters and should have the following properties:

- Every cluster should contain at least one data point, i.e . $C_i \neq \phi \forall i \in \{1, 2, 3, \ldots, k\}$
- Each data point must belong to a cluster i.e $U_{i=1}^k C_i = O$
- No data point should belong to more than one cluster i.e. $C_i = \phi, \forall i \neq$ and $i, j \in \{1, 2, 3, \ldots, k\}$.

A dataset is clustered when the above conditions get satisfied but quality clustering is done when clusters satisfy the fitness condition. The most widely used fitness condition to specify the quality of the clustering (Yang et al. 2010) is the total mean square quantization error(MSE) which is defined by Eq. (7).

$$f(O, C) = \sum_{l=1}^{k} \sum_{O_i \epsilon C_l} d(O_i, Z_l)^2 \tag{7}$$

where $d(o_i, z_l)$ is the measure of dissimilarity between the data point o_i and centroid of the cluster $c_l(z_l)$ which is the mean value of the data points belonging to the respective cluster. Dissimilarity between the clusters is calculated by different distance metric. The most common distance metric for calculating the dissimilarity is the Euclidean distance which is defined by Eq. (8).

$$(X_i, X_j) = \sqrt{\sum_{p=1}^{d} (x_i^p - x_i^p)^2} \tag{8}$$

3.3 k-mean algorithm

k-mean algorithm (Jain 2010; Forgy 1965; Kaufman and Rousseeuw 2009) is one of the popular clustering algorithm present in the literature that has been widely applied to many applications. It starts with random cluster centroids and then each data point is assigned to closest cluster based on its distance from the centroids. Centroids are updated with iterations by calculating the mean of the data points belonging to a particular cluster. Algorithm terminates either after completing the maximum number of iterations or the same cluster centroids repeat.

4 Proposed approach

In this section the proposed variant of the Bat algorithm along with its hybridization with k-means algorithm has been explained. The idea of the proposed DFBPKBA is to use the strength of bat algorithm for obtaining the global best solution and k-means algorithm for better population initialization. The clustering process of the proposed algorithm has been explained in the Sect. 4.1. Section 4.2 contains the parallelization of the proposed approach for the clustering using the hadoop and MapReduce.

4.1 Dynamic frequency based k-bat clustering

In the proposed DFBPKBA, two main operations are employed for performing the clustering task on large data sets. The first is, evaluation of the fitness value in the form of sum of square of mean Euclidean distance between each data object and centroid of the cluster. Secondly the cluster centroids are updated using the original bat algorithm described in Sect. 3.1 with dynamic frequency. The $freq_{max}$ parameter of the original bat algorithm is fix while the proposed DFBPKBA uses dynamic $freq_{max}$ value ,which

keep on decreasing with iterations using Eq. (5) that makes a good initial exploration followed by the exploitation. The fitness of each bat is calculated using the Eq. (8). The goal is to minimize the fitness value of each bat i.e sum of mean Euclidean distance. Further the quality of the clusters and convergence speed of the dynamic frequency based Bat algorithm has been enhanced by incorporating k-means algorithm. It is important to create a good initial population because the performance of the Bat algorithm and most of the meta-heuristic algorithm is much effected by the initial population.

$$freq_{max}(i) = (freq_{max} - itr_i * (freq_{max}/maxit)) \quad (9)$$

In Eq. (9) $freq_{max}$ is the initial maximum frequency, $freq_{max}(i)$ is the current maximum frequency, itr_i is the current iteration number and $maxit$ is the maximum number of iterations for which the algorithm is run. The clustering process using the dynamic frequency based bat algorithm is based on the three main steps.

In the first step, ten iterations of k-means algorithm is applied to the dataset to get some nearby solution and also to avoid local optima. In the second step the population is initialized by the centroids produced by the ten iterations of the k-means algorithm. In the third step each data point is assigned a cluster using dynamic frequency based bat algorithm. The candidate solution in the proposed algorithm is represented by the one dimensional array to encode the centroids of the cluster centers. The size of the array is a*k where a is the number of the attributes of the data set and k is the number of predefined cluster. Let us consider $Pi = \{C_1, C_2, C_3 \ldots C_k\}$ as the i_{th} candidate solution, then $C_j = \{C_j^1, C_j^2, C_j^3, C_j^a\}$ is the Jth cluster centroid of the Ith candidate solution $(i = 1, 2, 3 \ldots N)$ and $j(1, 2, 3 \ldots a)$. N is the number of candidate solution or the population size of the algorithm. Table 1 contains population size taken for performing the clustering using the proposed DFBPKBA along with the other parameter values.

Table 1 Parameter values

Parameter name	BAT	PSO	DFBPKBA
Population size (pop)	40	40	40
$freq_{min}$	0	–	0
$freq_{max}$	10	–	10
alpha	0.9	–	.9
gamma	0.01	–	0.01
r0	0.5	–	0.5
Inertia weight	–	0.5	–
C1	–	0.5	–
C2	–	0.5	–

4.2 Parallelization of the proposed algorithm using MapReduce architecture

For the parallel processing to handle large scale data we have used Hadoop MapReduce model. Hadoop Shvachko et al. (2010) is an open source platform for distributed processing of large scale data sets. Hadoop works on the principle of distributing the large amount of work among the various machines to achieve speedup. The main reason for its popularity is its efficient task scheduling, concurrency control, robustness, scalability, and node failure recovery. MapReduce (Dean and Ghemawat 2008) is a parallel processing paradigm which have capability to process large scale data on the cluster of commodity hardware. In this programming model the large data set is broken into smaller chunks and distributed among various machines. Each machine have some part of the original data upon which map reduce job is rum independently. MapReduce framework process the data in the form of key, value pair. The whole work is done in two phases namely Map and Reduce phase. Map function is invoked for each input key,value pair and it generates intermediate output in the form of key,value. In the second phase reduce function is called for each key,value pair which was generated by the mappers of the first phase.The reduce function processes the key,value pair and produces the final output. For the clustering of large scale data set the main computation intensive task is to compute the fitness function since it involves visiting of each data object. Since in the MapReduce based approach each machine have a smaller part of the data so parallelism is achieved resulting in fast processing. Such approach becomes more important for the huge size data but when the data size is less, in such cases speedup may be reduced because of the hidden input output cost. As a whole, the procedure of DFBPKBA is described in Fig. 1.

4.2.1 Mapper

The job of each mapper is to calculate the distance from each data object from each cluster center. Thus map function finds the distance of data object from each centroid and compares those distances for one agent and one data point. After the calculation of the minimum distance of a data point from the centroids of the cluster, the mapper emits the batID and the computed minimum distance value as an output. Map process is thus used to compute the fitness of each bat. So the input to the mapper will be the data object and output will be the sum of the minimum distances of the data objects from the centroid. The map function is shown in Algorithm 1.

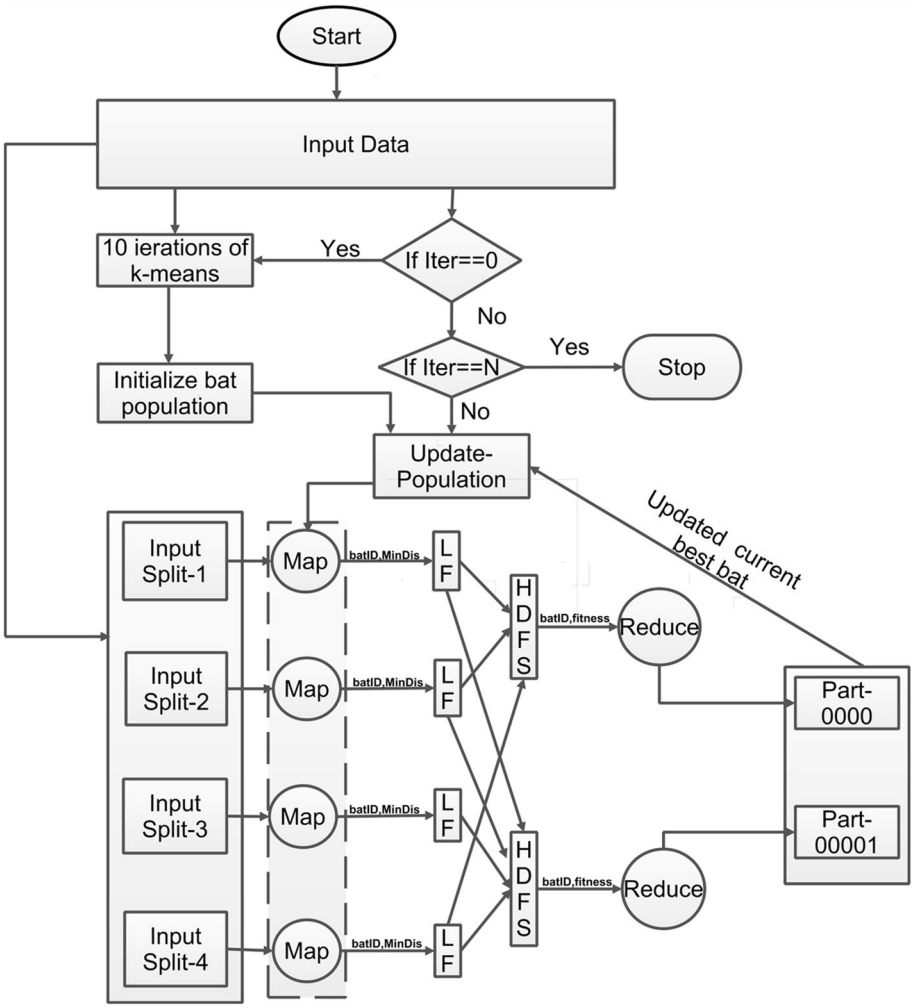

Fig. 1 The procedure of DFBPKBA based on MapReduce

Algorithm 1 Map Function

Input: $Key - data, Value - Dataobject.$
Output: .Key-BatId, Value-Minimum distance
Map (Key : dataId, Value : data)
Bat-list = read (file);
For each bat;
$Minimun - distance$= getMinDistance (bat, data);
$Min - Distance$=$Minimun - distance + centroid - index$
write (badID, Min-Distance);

4.2.2 Reduce

Reduce function starts with retrieving the batID and its fitness value i.e Min-Distance that is produced by the Map phase. The output key, value pair produced by the Map phase are sorted and grouped by key by the Hadoop. In our module bat-id is the key, therefore the values with the same batID will be grouped together. Finally the reducer will club the outputs got from the various mappers and emits the key with sum of mean squared Euclidean distance as value. The sum of mean squared Euclidean distance as value is the total intra-cluster distance that we aim to minimize. The pseudo code of this reducer module is given in Algorithm 2.

Algorithm 2 Reduce Function

Input: $Key - batId, Value - Min - Distance$
Output: .Key-Bat, Value-bat fitness
Reduce (Key: batID, Value: Min)
For each dist in Min-distance list
fitness=fitness+dist
if ($newfitness > oldfitness$)
$updateoldfitness = newfitness$
update position vector
decrease loudness
increase pulse rate
writeToFile (batID, fitness)

After mappers and reducers have finished working, the combine module is used to find the best bat. After getting the best bat, the population is updated in the driver code for the next iteration.

Table 2 Simulation results for the clustering algorithm

Dataset name	Criteria	K-means	PSO	BAT	DFBPKBA
Iris	Best	97.5674	96.9087	104.786	**96.5555**
	Average	105.129	98.8976	118.9807	**96.6767**
Glass	Best	214.5467	223.7685	341.3211	**198.8769**
	Average	227.9778	230.49328	380.9874	**201.7654**
Wine	Best	16,566.77	**16,304.48**	16,768.66	16,396.03
	Average	16,963.05	**16,316.27**	17,094.89	16,461.38
Magic	Best	1,650,422.68	1,659,260.50	2,205,689.82	**1,645,851.75**
	Average	1,660,311.11	1,659,210.50	2,635,125.55	**1,647,410.30**
Pokerhand	Best	6,652,657.44	6,750,253.48	6,675,069.58	**6,031,523.30**
	Average	6,669,935.77	6,887,355.11	6,698,433.00	**6,055,335.71**

The results (best and average) are represented in bold font

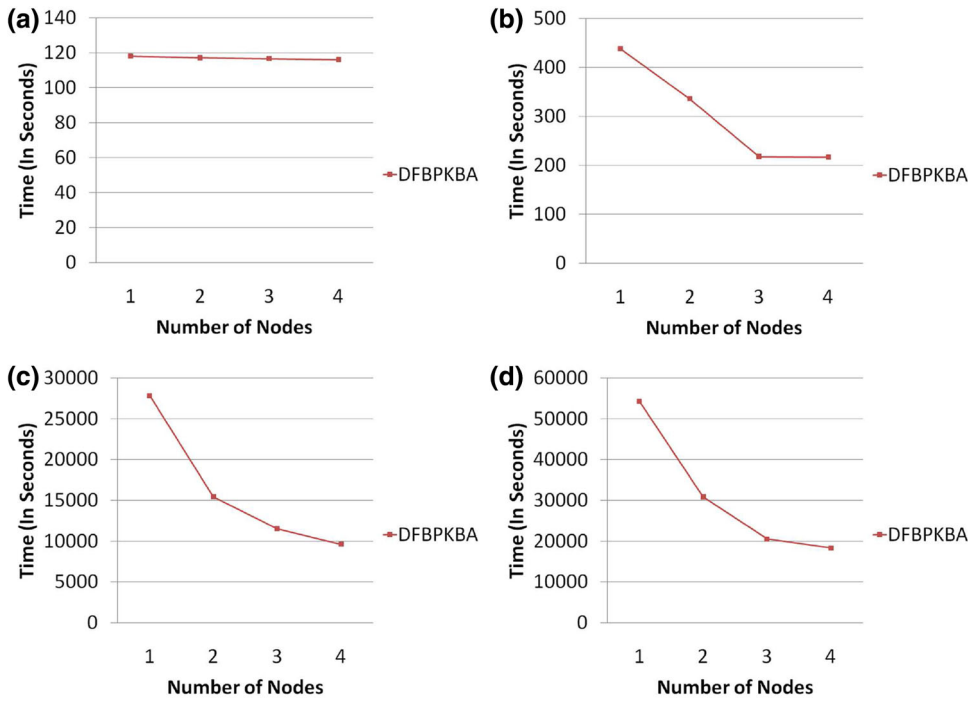

Fig. 2 The speedup graph of **a** wine **b** magic **c** Pokerhand **d** replicated wine

5 Experimental results

The proposed algorithm is tested on five benchmark datasets and compared with three algorithms, namely k-mean, particle swarm optimization (PSO) based clustering and original bat clustering. For the fair comparison, all the algorithms were run with 500 iterations and parameters setting of each algorithm is given in the Table 1. The performance of the proposed algorithms have been evaluated in terms of clustering quality and speedup. The DFBPKA is run on the cluster of 4 nodes having Intel Corei5-4570 processors with processing 3.20 GHz 4.32-bit configuration, Ubuntu 14.04 with Hadoop version 2.7.2. of replication value as 3. All the machines have the same configurations and same version of the Hadoop. The benchmark datasets are taken from UCI machine learning repository (Blake and Merz 1998). All these datasets are widely used to validate the efficiency of the algorithms for clustering.

5.1 Cluster quality

The cluster quality is measured with two parameters i.e best and average intra-cluster distance (Jain 2010; Hatamlou et al. 2012). Each algorithm is run thirty time on all the datasets. Table 2 contains the best and average intra-cluster distance of five datasets over thirty runs. It can be observed from the Table 2 that the proposed DFBPKBA has outperformed K-Means, PSO, BAT algorithm on the four datasets out of five in the terms of mean and average

Table 3 Speedup of the proposed algorithm on various data sets

Dataset name	No of records	Speedup on 2 nodes	Speedup on 3 nodes	Speedup on 4 nodes
Wine	178	1.008547009	1.012875536	1.017241379
Magic	19,020	1.305125149	2.011019284	2.023094688
PokerHand	1,000,000	1.800334185	2.41022842	2.887202766
Replicated wine	1,780,000	1.757780133	2.637304901	2.95502798

intra-cluster value. However for one data i.e Wine PSO has outperformed other algorithm for the mean and average value of intra-cluster distance.

5.2 Speedup test

The speedup of the proposed algorithm is tested on different sizes of the datasets. Two small size datasets are considered namely, Wine and Magic and two large size data sets are taken namely, Pokerhand and replicated wine which is made by replicating each record of the wine several time. In the experiment, the speedup of algorithm is calculated by Eq. (10).

$$Speedup = T_{base}/T_N \tag{10}$$

where T_{base} and T_N are the running time chosen as the baseline for the comparison and the running time of the system with N number of nodes, respectively. It is validated from the Fig. 2 that the proposed algorithm achieved good speedup on large scale data sets. It can also be depicted from the Fig. 2b–d that the running time of the DFBPKA decreases almost linearly with increasing number of nodes. However as we can see from the Fig. 2a that for the smaller data sets, the speedup performance is dropped because of the hidden input output cost. Table 3 shows the data sets taken and the corresponding speedup achieved by the proposed algorithm.

6 Conclusion

In this paper, a novel dynamic frequency based parallel bat algorithm (DFBPKBA), is proposed. The proposed algorithm takes the advantage of the bat algorithm to achieve global optimal solution and MapReduce architecture to handle large scale datasets. The DFBPKBA changes the max frequency parameter of original bat algorithm at each iteration to leverage the exploration at the starting stage followed by the exploitation at the later stage. The results shows that the proposed algorithm has outperformed PSO, k-means and Bat algorithm in the terms of quality of the clustering. The speedup performance results show that the DFBPKBA algorithm is able to handle large scale datasets efficiently in reasonable amount of time. We can conclude that the proposed DFBPKBA algorithm can serve as an useful alternative for clustering massive scale datasets. The future work includes the applicability of the proposed algorithm on real-world datasets of different domains like image or video.

References

Aljarah I, Ludwig SA (2013) Towards a scalable intrusion detection system based on parallel pso clustering using mapreduce. In: Proceedings of the 15th annual conference companion on Genetic and evolutionary computation, ACM, pp 169–170

Bansal JC, Sharma H, Jadon SS, Clerc M (2014) Spider monkey optimization algorithm for numerical optimization. Memet Comput 6(1):31–47

Bhavani R, Sadasivam GS, Kumaran R (2011) A novel parallel hybrid k-means-de-aco clustering approach for genomic clustering using mapreduce. In: 2011 world congress on information and communication technologies (WICT), IEEE, pp 132–137

Blake C, Merz CJ (1998) UCI repository of machine learning databases. Department of Information and Computer Science, Irvine

Cai S-J, Tsai P-W (2016) Echolocation guided evolved bat algorithm. J Inf Hiding Multimed Signal Process 7(1):153–162

Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113

del Río S, López V, Benítez JM, Herrera F (2014) On the use of mapreduce for imbalanced big data using random forest. Inf Sci 285:112–137

Fayyad UM, Wierse A, Grinstein GG (2002) Information visualization in data mining and knowledge discovery. Morgan Kaufmann, Burlington

Forgy EW (1965) Cluster analysis of multivariate data: efficiency versus interpretability of classifications. Biometrics 21:768–769

Frontpage–hadoop wiki. http://wiki.apache.org/hadoop/, (Accessed on 09/17/2016)

Gong Y-J, Chen W-N, Zhan Z-H, Zhang J, Li Y, Zhang Q, Li J-J (2015) Distributed evolutionary algorithms and their models: a survey of the state-of-the-art. Appl Soft Comput 34:286–300

Hatamlou A, Abdullah S, Nezamabadi-Pour H (2012) A combined approach for clustering based on k-means and gravitational search algorithms. Swarm Evolut Comput 6:47–52

Jadon SS, Bansal JC, Tiwari R, Sharma H (2014) Artificial bee colony algorithm with global and local neighborhoods. Int J Syst Assur Eng Manag 1–13

Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recogn Lett 31(8):651–666

Jansen BJ, Zhang M, Sobel K, Chowdury A (2009) Twitter power: tweets as electronic word of mouth. J Am Soc Inform Sci Technol 60(11):2169–2188

Kaufman L, Rousseeuw PJ (2009) Finding groups in data: an introduction to cluster analysis. Wiley, New York

Khezr SN, Navimipour NJ (2015) Mapreduce and its application in optimization algorithms: a comprehensive study. Majlesi J Multimed Process 4(3)

Lin K-Y, Xu L-H, Wu J-H (2004) A fast fuzzy c-means clustering for color image segmentation. J Image Gr 2:005

Lin C-Y, Pai Y-M, Tsai K-H, Wen CH-P, Wang L-C (2013) Parallelizing modified cuckoo search on mapreduce architecture. J Electr Sci Technol 11(2):115–123

Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks. In: IJCAI, pp 3818–3824

Meena MJ, Chandran K, Karthik A, Samuel AV (2012) An enhanced aco algorithm to select features for text categorization and its parallelization. Expert Syst Appl 39(5):5861–5871

Moertini VS, Venica L (2016) Enhancing parallel k-means using map reduce for discovering knowledge from big data. In: 2016 IEEE international conference on cloud computing and big data analysis (ICCCBDA), IEEE, pp 81–87

Nguyen T, Pan J, Chu S, Roddick JF, Dao TK (2016) Optimization localization in wireless sensor network based on multi-objective firefly algorithm. J Netw Intell 1(4):130–138

Sharma K, Chhamunya V, Gupta P, Sharma H, Bansal JC (2015) Fitness based particle swarm optimization. Int J Syst Assur Eng Manag 6(3):319–329

Shvachko K, Kuang H, Radia S, Chansler R (2010) The hadoop distributed file system. In: 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE, 1–10

Tsai P-W, Zhang J, Zhang S, Istanda V, Liao L-C, Pan J-S (2015) Improving swarm intelligence accuracy with cosine functions for evolved bat algorithm. J Inf Hiding Multimed Signal Process 6:1194–1202

Tsai PW, Zhang J, Liu Y, He Y, Zhang S, Pan J-S (2016) Undulating swarm intelligence agents in wave increasing evolved bat algorithm. J Inf Hiding Multimed Signal Process 7(1):21–30

Verma A, Llorà X, Goldberg DE, Campbell RH (2009) Scaling genetic algorithms using mapreduce. In: 2009 ninth international conference on intelligent systems design and applications, IEEE, pp 13–18

Wang J, Yuan D, Jiang M (2012) Parallel k-pso based on mapreduce. In: 2012 IEEE 14th international conference on communication technology (ICCT), IEEE, pp 1203–1208

Wu B, Wu G, Yang M (2012) A mapreduce based ant colony optimization approach to combinatorial optimization problems. In: 2012 eighth international conference on natural computation (ICNC), IEEE, pp 728–732

Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16:645–678

Xu X, Ji Z, Yuan F, Liu X (2014) A novel parallel approach of cuckoo search using mapreduce. In: 2014 international conference on computer, communications and information technology (CCIT 2014), Atlantis Press

Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010), Springer, pp 65–74

Yang X-S, He X (2013) Bat algorithm: literature review and applications. Int J Bio-Inspired Comput 5(3):141–149

Yang S, Wu R, Wang M, Jiao L (2010) Evolutionary clustering based vector quantization and spiht coding for image compression. Pattern Recogn Lett 31(13):1773–1780

You Z-H, Yu J-Z, Zhu L, Li S, Wen Z-K (2014) A mapreduce based parallel svm for large-scale predicting protein-protein interactions. Neurocomputing 145:37–43