ISSN: 2229-6956 (ONLINE)
DOI: 10.21917/ijsc.2015.0151

MBA-LF: A NEW DATA CLUSTERING METHOD USING MODIFIED BAT ALGORITHM AND LEVY FLIGHT

R. Jensi[1] and G. Wiselin Jiji[2]
Department of Computer Science and Engineering, Dr. Sivanthi Aditanar College of Engineering, India
E-mail: [1]r_jensi@yahoo.co.in, [2]jijivevin@yahoo.co.in

Abstract

Data clustering plays an important role in partitioning the large set of data objects into known/unknown number of groups or clusters so that the objects in each cluster are having high degree of similarity while objects in different clusters are dissimilar to each other. Recently a number of data clustering methods are explored by using traditional methods as well as nature inspired swarm intelligence algorithms. In this paper, a new data clustering method using modified bat algorithm is presented. The experimental results show that the proposed algorithm is suitable for data clustering in an efficient and robust way.

Keywords:

Data Clustering, Bat Algorithm, Levy Flight, Global Optimization

1. INTRODUCTION

Cluster analysis or clustering [1],[2] is the process of grouping a set of objects into clusters/groups so that objects in the same group are having high degree of similarity based on some criteria while the objects in other groups are dissimilar. Data clustering is widely used in many areas including data mining, statistical data analysis, machine learning, pattern recognition, image analysis, information retrieval and etc. As [2], clustering methods can be categorized into partitional methods, hierarchical methods, density-based methods, grid-based methods and model-based methods.

Among the several clustering methods, partitional clustering methods are heavily used. K-means algorithm is one of the partitional and center-based clustering algorithms [6]. Owing to the initialization of cluster centers, k-means clustering algorithm traps into local optima.

Over the last few decades, many nature-inspired evolutionary algorithms are being developed for solving most engineering design optimization problems. Nature-inspired algorithms [7] [8] mimic the behaviours of the living things in the nature, so they are also called as Swarm Intelligence (SI) algorithms. SI algorithms searches for global optima and also has good convergence speed.

Evolutionary algorithms (EAs) were the initial stage of such optimization methods [9]. Genetic Algorithm (GA) [10] and Simulated Annealing (SA) [11] are popular examples for EAs. In the early 1970s, Genetic algorithm was developed by John Holland, which inspired by biological evolution such as reproduction, mutation, crossover and selection. Simulated annealing (SA) was developed from inspiration by annealing in metallurgy, a technique involving heating and cooling of a material to increase the size of its crystals and reduce their defects.

The nature inspired algorithms include Particle Swarm Optimization (PSO) [12] [13], Ant Colony Optimization (ACO) [14], Glowworm Swarm Optimization (GSO) [15], Bacterial Foraging Optimization (BFO) [16]-[17], the Bees Algorithm [18], Artificial Bee Colony algorithm (ABC) [19]-[21], Biogeography-based optimization (BBO) [22] , Cuckoo Search (CS) [23]-[24], Firefly Algorithm (FA) [25]-[26] , Bat Algorithm (BA) [27] ,flower pollination algorithm[28] and Krill herd algorithm [29].

Swarm Intelligence system holds a population of solutions, which are changed through random selection and alterations of these solutions. The way, the system differs depends on the generation of new solutions, random selection procedure and candidate solution encoding technique. Particle Swarm Optimization (PSO) was developed in 1995 by Kennedy and Eberhart simulating the social behaviour of bird flock or fish school. Ant Colony Optimization, introduced by Dorigo, imitates the food searching paths of ants in nature. Glowworm Swarm Optimization (GSO) was introduced by Krishnanand and Ghose in 2005 based on the behaviour of glow worms. Bacterial foraging optimization algorithm was developed based on the foraging behaviour of bacteria such as E.coli and M.xanthus. The Bees Algorithm was developed by Pham DT in 2005 imitating the food foraging behaviour of honey bee colonies. Artificial bee colony algorithm was developed by Karaboga, being motivated from food foraging behaviour of bee colonies. Biogeography-based optimization (BBO) was introduced in 2008 by Dan Simon inspired by biogeography, which is the study of the distribution of biological species through space and time. Cuckoo search was developed by Xin-she Yang and Subash Deb in 2009 being motivated by the brood parasitism of cuckoo species by laying their eggs in the nests of other host birds. Firefly algorithm was introduced by Xin-She Yang inspired by the flashing behaviour of fireflies. The primary principle for a firefly's flash is to act as an indicator system to draw other fireflies. Bat algorithm was developed in 2010 by Xin-She Yang based on the echolocation behaviour of microbats. Flower pollination algorithm was developed by Xin-She Yang in 2012 motivated by the pollination process of flowering plants. In 2012, Gandomi and Alavi was developed a new bio-inspired algorithm based on the simulation behaviour of krill species.

The remaining section of this paper is organized as follows. Section 2 presents some of the previous proposed research work on data clustering. Bat algorithm and proposed modified bat levy algorithm is presented in Section 3 and Section 4 respectively. Section 5 presents experimental results followed by conclusion in section 6.

2. RELATED WORK

Van, D.M. and A.P. Engelbrecht. (2003) [13] proposed data clustering approach using particle swarm optimization. The author proposed two approaches for data clustering. The first

approach is PSO, in which the optimal centroids are found and then these optimal centroids were used as a seed in K-means algorithm and the second approach is, the PSO was used to refine the clusters formed by K-means. The two approaches were tested and the results show that both PSO clustering techniques have much potential.

Ant Colony Optimization (ACO) method for clustering is presented by Shelokar et al. (2004) [14]. In [14], the authors employed distributed agents that imitate the way real-life ants find the shortest path from their nest to a food source and back. The results obtained by ACO can be considered viable and is an efficient heuristic to find near-optimal cluster representation for the clustering problem.

Kao et al. (2008) [31] proposed a hybridized approach that combines PSO technique, Nelder–Mead simplex search and the K-means algorithm. The performance of K-NM-PSO is compared with PSO, NM-PSO, K-PSO and K-means clustering and it is proved that K-NM-PSO is both strong and suitable for handling data clustering.

Selim and Al-Sultan (1991) [11] presented a simulated annealing approach to the data clustering problem and they proved that the algorithm obtained global optimum solution. Maulik and Bandyopadhyay (2000) [10] proposed a genetic algorithm approach to the clustering. The superiority of the GA-clustering algorithm over the K-means is demonstrated for synthetic and real-life datasets. Karaboga and Ozturk (2011) [35] presented a new clustering approach using Artificial Bee Colony (ABC) algorithm which simulates the food foraging behaviour of a honey bee swarm. The performance is compared with PSO and other classification techniques. The simulation results show that the ABC algorithm is superior to other algorithms.

Zhang et al. (2010) [32] presented the artificial bee colony (ABC) as a state-of-the-art approach to clustering. Deb's rules are used to tackle infeasible solutions instead of the greedy selection process usually used in the ABC algorithm. When they tested their algorithm, they found very encouraging results in terms of effectiveness and efficiency.

In [36] (2012), X. Yan et al presented a new data clustering algorithm using hybrid artificial bee colony (HABC). The genetic algorithm crossover operator was introduced to ABC to enhance the information exchange between bees. The HABC algorithm achieved better results.

Tunchan Cura. (2012) [33] presented a new PSO approach to the data clustering and the algorithm was tested using two synthetic datasets and five real datasets. The results show that the algorithm can be applied to clustering problem with known and unknown number of clusters. Senthilnath, J., Omkar, S.N. and Mani, V. (2011) [37] presented data clustering using firefly algorithm. They measured the performance of FA with respect to supervised clustering problem and the results show that algorithm is robust and efficient.

M.Wan and his co-authors (2012) [38] presented data clustering using Bacterial Foraging Optimization (BFO). The algorithm proposed by these researchers was tested on several well-known benchmark data sets and Compared three clustering technique. The author concludes that the algorithm is effective and can be used to handle data sets with various cluster sizes, densities and multiple dimensions.

J. Senthilnatha, Vipul Dasb, Omkara, V. Mani, (2012) [39] proposed a new data clustering approach using Cuckoo search with levy flight. Levy flight is heavy-tailed which ensures that it covers output domain efficiently. The author concluded that the proposed algorithm is better than GA and PSO.

3. BAT ALGORITHM

Bat algorithm [3] is a metaheuristic algorithm inspired from the echolation behaviour of bats. A population of n bats (solutions) is initialized $X = \{x_1, x_2, \ldots, x_n\}$ and for each solution x_i pulse frequency f_i is randomly initialized in the range $[f_{min}, f_{max}]$ and loudness A_i, pulse rate r_i are initialized.

The new solution and velocities for each solution at the next generation is found by,

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta \tag{1}$$

$$v_i^t = v_i^{t-1} + \left(x_i^t - x_{best}\right) f_i \tag{2}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{3}$$

where, β is a uniformly distributed random number between 0 and 1, x_{best} is the global best solution among all the solutions, f_i is the velocity increment.

The local search is performed with pulse rate probability. During the local search, new solution for each bat is generated using,

$$x_{new} = x_{old} + \in A^t \tag{4}$$

where, $\in$ is a random number in the range [-1,1], A^t is the average loudness of all the bats at the generation. In each generation, the loudness A_i and pulse rate r_i is updated according to,

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

$$r_i^{t+1} = r_i^0 * (1 - \exp(-\gamma t)) \tag{6}$$

where, α and γ are constants and it is assigned the value 0.9 [3].

A variant of bat algorithm is given in [4] and each algorithm is used for the purpose of global optimization. Komarasamy and Wahi (2012) [5] presented a data clustering technique using the combination of K-means and bat algorithm (KMBA). The bat algorithm converges quickly due to the switch over to exploitation stage by varying loudness *A* and pulse rate *r*, leads to stuck in local optima. In this study we present a modified bat algorithm combined with levy flight for efficient data clustering.

4. PROPOSED WORK

The proposed work aims to cluster the data objects efficiently and effectively. The modified bat algorithm is presented here, with which we can achieve the better clustering result. Levy flight is employed to explore the search space randomly. Levy flight or levy walk is stochastic walk in which step length is expressed by a 'heavy-tailed' probability distribution. Levy flight can explain all stochastic processes that are scale invariant. Levy walks are drawn from Levy stable distribution. This distribution is a simple power-law formula $L(s) \sim |s|^{-1-\beta}$ where $0 < \beta < 2$ is an index. Mathematically, a simple version of Levy distribution can be defined as [8], [20]:

$$L(s,\gamma,\mu)=\begin{cases}\sqrt{\frac{\gamma}{2\pi}}\exp\left[-\frac{\gamma}{2(s-\mu)}\right]\frac{1}{(s-\mu)^{3/2}} & \text{if } 0<\mu<s<\infty\\ 0 & \text{if } s\le 0\end{cases} \quad (7)$$

where, μ parameter is location or shift parameter, $\gamma > 0$ parameter is scale (controls the scale of distribution) parameter.

The fitness function used in this paper is the squared error function [30]:

$$f=\sum_{j=1}^{k}\sum_{i=1}^{N}\min\left(d\left(X_i,C_j\right)\right) \quad (8)$$

where, $d(X_i, C_j)$ is the dissimilarity measure between data object X_i and cluster center C_j.

There are several similarity/dissimilarity measurement exist including Euclidean distance, Cosine similarity, Manhattan distance, Jaccard coefficient, etc., [2]. In this paper, the popular Euclidean distance measure is used and it is defined for two objects X_i and X_j as follows [2]:

$$d\left(X_i,X_j\right)=\sqrt{\sum_{d=1}^{m}\left(x_i^d-x_j^d\right)^2} \quad (9)$$

where, m is the number of features/dimensions of the data object.

The proposed algorithm keeps a population of solutions and improves the solution at each iteration. The candidate solution is represented as a row vector of size $k \times m$ where, k is the number of clusters and m is the dimension of the each data object. It is shown in Fig. 1.

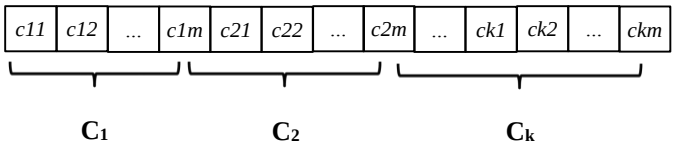

Fig.1. Example of a candidate solution for k clusters and m features

The Fig.1. shows a candidate solution representation in which C_1 represents the first cluster's centroid, C_2 represents the second cluster's centroid and C_k represents the k^{th} cluster centroid. So the population of solution is represented as,

$$P=\begin{bmatrix}S1'\\ S2'\\ S3'\\ \vdots\\ SN'\end{bmatrix} \quad (10)$$

where, Si' is a candidate solution, N is the population size.

To improve the performance of the proposed algorithm, crossover operator is applied. The crossover operation is controlled by crossover probability. In our study, a binomial crossover is employed [29], [34]. The m^{th} component of Xi, $x_{i,m}$ is generated as:

$$x_{i,m}=\begin{cases}x_{r,m}, & rand_{i,m}<C_r\\ x_{i,m}, & else\end{cases} \quad (11)$$

where, r is the randomly generated local solution around the best solution when rand > r_i.

The proposed algorithm is summarized as follows:

1. Initialize the parameters population size N, maximum number of generations N_gen, loudness A_i, pulse rate r_i, alpha, lamda, max, min frequency range. Set $t = 0$.
2. Randomly initialize a population of solutions.
3. Evaluate the fitness value and find the current global best.
4. While (t<N_gen)
   - 4.1 Store the best bat.
   - 4.2 For each solution i do
     - 4.2.1 Calculate f_i according to Eq.(1).
     - 4.2.2 Calculate pulse frequency according to Eq.(2).
     - 4.2.3 Compute new solution x_i using Eq.(3).
     - 4.2.4 Evaluate the solution and update it if the solution improves.
     - 4.2.5 If rand>r_i then
       Generate a local solution around the best solution using Eq.(4).
       Perform crossover using Eq.(11).
       End if
     - 4.2.6 Generate a new solution by using Levy Flight.
     - 4.2.7 Evaluate the newly generated solution $f(x_i')$.
     - 4.2.8 Accept new solutions if $f(x_i')<f(x_i)$.
     - 4.2.9 If rand<A_i then
       Increase r_i and reduce A_i.
       End if
   - 4.3 Arrange the solutions and find the current global best.
   - 4.4 Replace worst bat with the best bat.
   - 4.5 $t = t + 1$
5. Output the global best solution.

5. EXPERIMENTAL RESULTS

The K-Means, Particle Swarm Optimization (PSO), Original Bat (OBAT) and proposed algorithm (MBA-LF) are written in Matlab 8.3 and executed in a Windows 7 Professional OS environment using Intel i3, 2.30 GHz, 2 GB RAM. The parameter values used for original bat and proposed algorithm is shown in Table.1.

Ten datasets were used to test the performance of our proposed method. These datasets characteristics are shown in Table.2. All data sets except Art1 and Art2 are available from UCI machine learning laboratory [40] or available at [41].

Table.1. Parameter Settings

Parameter	Value
Max Generation	300
Population Size(N)	40
alpha	0.95
lambda	0.95

Maximum frequency (f_{max})	$k \times m$
Minimum frequency (f_{min})	0
Crossover Probability	0.5

Table.2. Test dataset descriptions

	# of features	# of classes	# of instances (size of each class)
Art1	2	4	600(150,150,150,150)
Art2	3	5	250(50,50,50,50,50)
Iris	4	3	150(50,50,50)
Thyroid	5	3	215(150,35,30)
Wine	13	3	178(59,71,48)
Cancer	9	2	683(444,239)
CMC	9	3	1473(629,333,511)
Glass	9	6	214(70,17,76,13,9,29)
Crude oil	5	3	56(7,11,38)
Liver Disorder	6	2	345(145,200)

5.1 DATASETS DESCRIPTION

The ten datasets used in this study is described as follows:

Let,

n is the total number data objects to be clustered,

m is the number of attributes for each data object and

c is the number of clusters to be partitioned to.

Artificial datasets Art1 and Art2 are drawn from Kao. et.al (2008).

Data set 1: Artificial dataset 1 (Art1) (*n*=600, *m*=2, *k*=4)

This dataset is drawn from four independent bivariate normal distributions according to $N2\left(\mu=\begin{pmatrix}\mu_{i1}\\ \mu_{i2}\end{pmatrix}, \sum=\begin{bmatrix}0.5 & 0.05\\ 0.05 & 0.5\end{bmatrix}\right)$, i = 1, 2, 3, 4, $\mu_{11} = \mu_{12} = -3$, $\mu_{21} = \mu_{22} = 0$, $\mu_{31} = \mu_{32} = 3$, $\mu_{41} = \mu_{42} = 6$, μ and Σ being mean vector and covariance matrix respectively. The data set is shown in Fig.2.

Data set 2: Artificial dataset 2 (Art2) (*n*=250, *m*=3, *k*=5)

This dataset consists of 3 classes, each class contains 50 samples. Each sample is characterized with 3 features. These samples are drawn from five independent uniform distributions with ranges of [85,100], [70, 85], [55, 70], [40, 55] and [25, 40]. The data distribution of this dataset is shown in Fig.3.

Data set 3: Iris data (*n*=150, *m*=4, *k*=3)

Iris dataset contains 150 instances with four attributes. These instances fall under three groups namely Iris-setosa, Iris Versicolour, and Iris Virginica which are a type of iris plants. Each group has an equal of 50 instances.

Data set 4: Thyroid gland data (*n*=215, *m*=5, *k*=3)

This dataset consists of 215 samples with five attributes. All data fall into three categories of human thyroid diseases, namely, normal, hypothyroidism and hyperthyroidism. Thyroid gland data consists of 6 attributes. The first attribute denotes the class attributes indicating 1 for normal, 2 for hyperthyroidism and 3 for hypothyroidism. The remaining five attributes are considered for clustering the data, namely the T3-resin uptake test, total Serum thyroxin as measured by the isotopic displacement method, total serum triiodothyronine as measured by radioimmuno assay, basal thyroid-stimulating hormone as measured by radioimmuno of 200 mg of thyrotropin releasing-hormone and the basal value. All attributes are continuous.

Data set 5: Wine data (*n*=178, *m*=13, *k*=3)

This dataset contains data that are taken from chemical analysis of 178 wines grown in the same region in Italy but resulting from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. The 13 attributes are , namely, alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavonoids, nonflavonoid phenols, proanthocyanins, color intensity, hue, OD280/OD315 of diluted wines and praline.

Data set 6: Cancer data (*n*=683, *m*=9, *k*=2)

This dataset consists of 683 samples arrived from Dr.Wolberg clinical cases. Each instance has one of two possible classes: benign or malignant. The nine attributes include Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitoses

Data set 7: Contraceptive Method Choice (CMC) data (*n*=1473, *m*=9, *k*=3)

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The objects are married women who either were not pregnant or did not know if they were at the time of interview. The problem involves predicting the choice of the current contraceptive method of a woman based on her demographic and socio-economic characteristics. This dataset contains 1473 objects with nine attributes and three clusters.

Data set 8: Glass data (*n*=214, *m*=9, *k*=6)

This dataset contains 214 objects with nine attributes, namely, refractive index, sodium, magnesium, aluminum, silicon, potassium, calcium, barium and iron. The data were sampled from six different types of glass: float processed building windows, non-float processed building windows, float-processed vehicle windows, containers, tableware and headlamps.

Data set 9: Crude oil data (*n*=56, *m*=5, *k*=3)

This dataset consists of 56 samples about crude oil; each has five attributes and fall into 3 groups.

Data set 10: Liver disorder data (*n*=345, *m*=6, *k*=2)

This dataset consists of 345 data samples about blood tests of individuals who are thought to be liver disorders that might arise from excessive alcohol consumption. Each sample is characterized with six features.

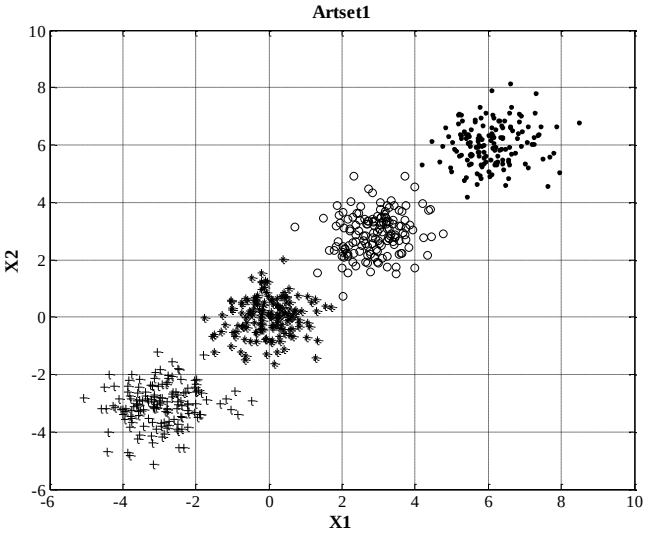

Fig.2. Artificial Dataset1 (Art1)

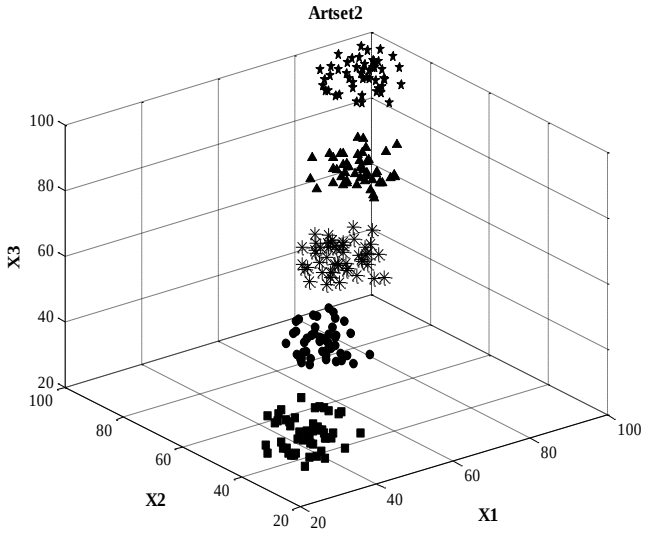

Fig.3. Artificial Dataset2 (Art2)

5.2 RESULTS AND DISCUSSION

In this paper, in order to compare the performance of proposed algorithm MBA-LF, K-means, original PSO algorithm and Original Bat algorithm are also executed and compared with proposed algorithm. Each algorithm was run 10 times and the results in Table 3-6 are mean, standard deviation, best and worst results of 10 experimental runs. All the three algorithms are executed on the ten datasets.

The clustering quality is measured based on the two criteria such as,

(i) Sum of intra-cluster distances: This is the sum of the distance between each data object in a cluster and centroid of that cluster which is computed by using Eq.(8). The smaller the distance is, the better the clustering result.

(ii) F-measure: This combines the precision and recall values used in information retrieval. The *precision* $P(i,j)$ and *recall* $R(i,j)$ for each class i of each cluster j are calculated as

$$P(i,j)=\frac{\gamma_{ij}}{\gamma_j} \tag{12}$$

$$R(i,j)=\frac{\gamma_{ij}}{\gamma_i} \tag{13}$$

where,

γ_i: is the number of members of class i

γ_j: is the number of members of cluster j

γ_{ij}: is the number of members of class i in cluster j

The corresponding *F-measure* $F(i,j)$ is given in Eq.(14):

$$F(i,j)=\frac{2*P(i,j)*R(i,j)}{P(i,j)+R(i,j)} \tag{14}$$

Then the definition of *F-measure* of a class i is given as,

$$F_{tot}=\sum_i \frac{\gamma_i}{n}\max_j\left(F(i,j)\right) \tag{15}$$

where, n is the total number of data objects in the collection. In general, the larger the F-measure gives the better clustering result.

The termination criteria used in the algorithm is the predefined maximum number of iterations.

Table.3. Intra-cluster distances for different algorithms for artificial datasets

Data set	Criteria	K-means	PSO	OBAT	MBA-LF
Art1	Average (Std)	531.529 (0.000)	531.258 (0.640)	565.157 (22.668)	530.874 (0.000)
	Best	531.529	530.874	537.113	530.874
	Worst	531.529	532.703	613.604	530.874
Art2	Average (Std)	2105.885 (398.983)	1769.001 (35.659)	2050.584 (189.291)	1727.153 (0.000)
	Best	1728.798	1729.668	1871.137	1727.153
	Worst	2508.766	1843.157	2486.510	1727.154

Table.4. Intra-cluster distances for different algorithms for real life datasets

Data set	Criteria	K-means	PSO	OBAT	MBA-LF
Iris	Average (Std)	99.847 (7.952)	97.057 (0.311)	103.036 (3.410)	96.657 (0.001)

	Best	97.326	96.669	97.433	96.656
	Worst	122.479	97.708	108.870	96.660
Thyroid	Average (Std)	1999.138 (12.005)	1883.378 (16.479)	1938.108 (27.795)	1872.289 (9.500)
	Best	1978.333	1868.747	1901.868	1866.467
	Worst	2017.046	1909.057	1974.214	1890.209
Cancer	Average (Std)	2987.988 (0.708)	2990.329 (16.416)	3107.125 (77.110)	2964.407 (0.008)
	Best	2986.961	2976.297	3021.483	2964.396
	Worst	2988.428	3023.612	3250.525	2964.419
Wine	Average (Std)	16743.807 (594.911)	16316.493 (14.300)	16606.901 (237.740)	16293.379 (0.944)
	Best	16555.679	16301.376	16391.462	16292.186
	Worst	18436.952	16337.523	17160.392	16294.390
CMC	Average (Std)	5543.128 (1.522)	5641.058 (45.034)	5802.144 (88.219)	5532.377 (0.134)
	Best	5542.182	5560.723	5671.526	5532.217
	Worst	5545.333	5702.539	5966.190	5532.610
Glass	Average (Std)	226.107 (15.662)	227.888 (4.701)	241.916 (5.059)	215.979 (2.272)
	Best	218.303	223.027	232.007	212.293
	Worst	260.772	239.243	247.085	219.917
Crude oil	Average (Std)	279.670 (0.162)	278.248 (0.707)	293.157 (5.957)	277.291 (0.028)
	Best	279.271	277.382	283.176	277.211
	Worst	279.743	279.582	306.670	277.302
Liver Disorder	Average (Std)	10216.327 (7.963)	9874.797 (18.804)	9973.219 (71.295)	9851.829 (0.173)
	Best	10212.549	9852.867	9876.685	9851.721
	Worst	10231.436	9912.223	10110.813	9852.080

The Table.3 lists the average, standard deviation, the best and worst intra-cluster distances obtained from the four algorithms for the two artificial datasets. From their results, it can be seen that the proposed MBA-LF algorithm outperforms PSO, OBAT and K-means in both artificial datasets. For Art1, the proposed algorithm always obtains the optimal intra-cluster distances in each run. For Art2, MBA-LF obtains more optimal intra-cluster distances than K-means, OBAT and PSO. The Table.5 lists the corresponding F-measure values for the two artificial datasets. For Art1, F-measure value is almost same for all the four algorithms and for Art2, PSO and MBA-LF achieves the maximum F-measure of 1 in all runs while K-means and OBAT algorithms obtain maximum F-measure for the best solution.

The Table.4 lists the average, standard deviation, the best and worst intra-cluster distances and Table.6 illustrates the corresponding F-measures from the four algorithms for the eight real life datasets. As can be seen clearly, MBA-LF outperforms well in terms of achieving optimal intra-cluster distances and higher F-measure. The proposed MBA-LF performs well those of compared with original bat algorithm. The reason for better performance of MBA-LF than OBAT is the high exploration of the solution space using random walk phenomenon of levy flight method. The F-measure value for CMC, Thyroid and Liver disorders datasets for the k-means are higher than PSO and MBA-LF.

The Fig.4-6 show the convergence behavior of artificial dataset Art1, Iris (low dimensional) and wine (high dimensional) datasets. It can be seen that, K-means algorithm quickly converges into local optima and furthermore there is no more improvement in searching. Thus K-means algorithm does not get optimal results.

On the other hand, PSO escapes from local optima and explores for better optimal solutions than K-means. When compared to K-means and PSO, MBA-LF algorithm escapes from local optima similar to PSO, but finds the better optimal solutions by searching randomly with less number of iterations. It reveals that MBA-LF algorithm explores the search space randomly and has fast convergence within 200 iterations with the help of Levy flight operation.

Table.5. F-measure values for different algorithms for artificial datasets

Data set	**Criteria**	**K-means**	**PSO**	**OBAT**	**MBA-LF**
Art1	Average (Std)	0.997 (0.000)	0.997 (0.000)	0.996 (0.001)	0.997 (0.000)
	Best	0.997	0.997	0.997	0.997
	Worst	0.997	0.997	0.995	0.997
Art2	Average (Std)	0.901 (0.104)	1.000 (0.000)	0.980 (0.062)	1.000 (0.000)
	Best	1.000	1.000	1.000	1.000
	Worst	0.790	1.000	0.804	1.000

Table.6. F-measure values for different algorithms for real life datasets

Data set	**Criteria**	**K-means**	**PSO**	**OBAT**	**MBA-LF**
Iris	Average (Std)	0.868 (0.069)	0.901 (0.011)	0.890 (0.026)	0.899 (0.000)

	Best	0.892	0.920	0.926	0.899
	Worst	0.670	0.892	0.832	0.899
Thyroid	Average (Std)	0.776 (0.096)	0.657 (0.023)	0.625 (0.030)	0.658 (0.029)
	Best	0.862	0.690	0.679	0.695
	Worst	0.652	0.619	0.588	0.619
Cancer	Average (Std)	0.961 (0.001)	0.964 (0.002)	0.958 (0.006)	0.965 (0.000)
	Best	0.962	0.966	0.966	0.965
	Worst	0.960	0.959	0.950	0.965
Wine	Average (Std)	0.707 (0.025)	0.723 (0.005)	0.720 (0.008)	0.723 (0.005)
	Best	0.715	0.729	0.729	0.729
	Worst	0.636	0.719	0.703	0.719
CMC	Average (Std)	0.403 (0.002)	0.402 (0.002)	0.405 (0.004)	0.402 (0.001)
	Best	0.406	0.407	0.414	0.403
	Worst	0.402	0.401	0.401	0.401
Glass	Average (Std)	0.519 (0.034)	0.512 (0.028)	0.505 (0.032)	0.530 (0.026)
	Best	0.557	0.553	0.551	0.555
	Worst	0.456	0.477	0.466	0.470
Crude oil	Average (Std)	0.666 (0.020)	0.700 (0.018)	0.701 (0.023)	0.707 (0.016)
	Best	0.722	0.722	0.737	0.722
	Worst	0.658	0.658	0.671	0.689
Liver Disorder	Average (Std)	0.646 (0.000)	0.623 (0.001)	0.620 (0.004)	0.623 (0.001)
	Best	0.646	0.624	0.626	0.624
	Worst	0.646	0.622	0.612	0.622

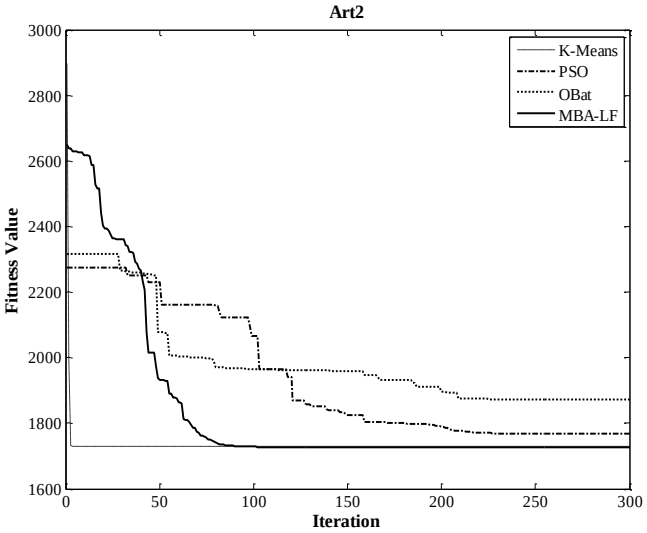

Fig.4. Convergence behavior of art2 dataset

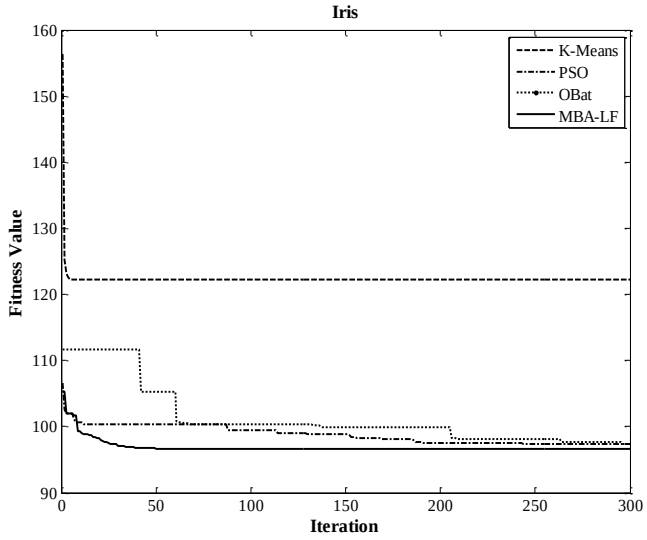

Fig.5. Convergence behavior of Iris dataset (low dimensional)

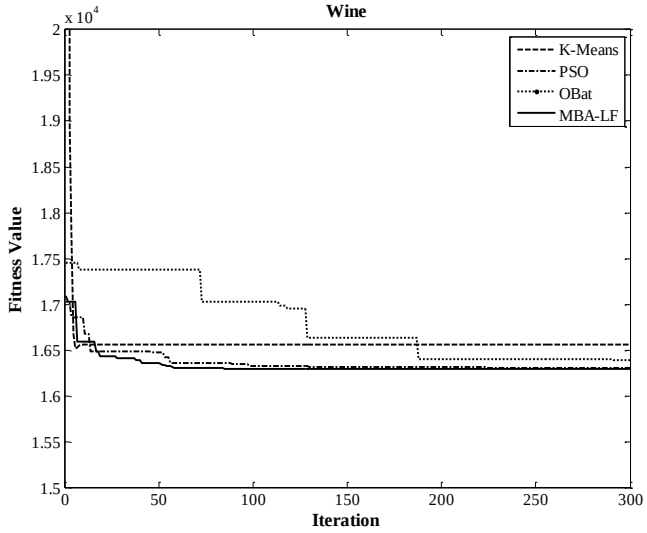

Fig.6. Convergence behavior of wine dataset (high dimesional)

6. CONCLUSION

In this paper, Levy flight is combined with modified Bat algorithm to improve the clustering result. The proposed approach is tested on ten datasets and the experimental results show that the proposed algorithm clusters the data objects efficiently. It also illustrates that it escapes from local optima and explores the search space effectively. In future, this work can be applied to other applications, for example text document clustering to cluster the set of documents efficiently.

REFERENCES

[1] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, "*Introduction to Data Mining*", Addison-Wesley Longman Publishing Co., Inc. Boston, 2005.

[2] Jiawei Han, Michelin Kamber and Jian Pei, "*Data Mining Concepts and Techniques*", Morgan Kaufmann, 2011.

[3] Xin-She Yang, "A New Metaheuristic Bat-Inspired Algorithm", *Nature Inspired Cooperative Strategies for Optimization*, *Studies in Computational Intelligence*, Vol. 284, pp. 65-74, 2010.

[4] Xin-She Yang, "Bat Algorithm: Literature Review and Applications", *International Journal Bio-Inspired Computation*, Vol. 5, No. 3, pp. 141-149, 2013.

[5] Komarasamy G and Amitabh Wahi, "An Optimized K-means Clustering Technique using Bat Algorithm", *European Journal of Scientific Research*, Vol. 84, No. 2, pp. 263-273, 2012.

[6] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman and Angela Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 881-892, 2002.

[7] James Kennedy, Russell C. Eberhart and Yuhui Shi, "*Swarm Intelligence*", Morgan Kaufmann, 2001.

[8] Xin-She Yang, "*Nature-inspired Metaheuristic Algorithms*", Luniver Press, 2010.

[9] R. Jensi and G. Wiselin Jiji, "A Survey on Optimization approaches to Text Document Clustering", *International Journal on Computational Science and Applications*, Vol. 3, No. 6, pp. 31-44, 2013.

[10] Ujjwal Malik and Sanghamitra Bandyopadhyay, "Genetic Algorithm-based Clustering Technique", *Pattern Recognition*, Vol. 33, No. 9, pp. 1455-1465, 2000

[11] Shokri Z. Selim and K. Al-Sultan, "A simulated annealing algorithm for the clustering problem", *Pattern Recognition*, Vol. 24, No. 10, pp. 1003-1008, 1991.

[12] Eberhart J and Eberhart R, "Particle Swarm Optimization", *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 1995.

[13] Van D.M. and A.P. Engelbrecht, "Data clustering using particle swarm optimization", *Proceedings of The Congress on Evolutionary Computation*, pp. 215-220, 2003.

[14] P.S. Shelokar, V.K. Jayaraman and B.D. Kulkarni, "An Ant Colony Approach for Clustering", *Analytica Chimica Acta*, Vol. 509, No. 2, pp. 187-195, 2004.

[15] K.N. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions", *Swarm Intelligence*, Vol. 3, No. 2, pp. 87-124, 2009.

[16] Swagatam Das, Arijit Biswas, Sambarta Dasgupta and Ajith Abraham, "Bacterial Foraging optimization Algorithm: Theoretical Foundations, Analysis, and Applications", *Foundations of Computational Intelligence*, Vol. 203, pp. 23-55, 2009.

[17] N.K. Jhankal and D. Adhyaru, "Bacterial foraging optimization algorithm: A derivative free technique", *Nirma University International Conference on Engineering*, pp.1-4, 2011.

[18] D.T. Pham and M. Castellani, "The Bees Algorithm – Modelling Foraging Behaviour to Solve Continuous Optimization Problems", *Proceedings of the Institution of Mechanical Engineers Part C: Journal of Mechanical Engineering Science*, Vol. 223, No. 12, pp. 2919-2938, 2009.

[19] Dervis Karaboga and Bahriye Akay, "A comparative study of Artificial Bee Colony algorithm", *Applied Mathematics and Computation*, Vol. 214, No. 1, pp. 108-132, 2009.

[20] Barthelemy P, Bertolotti J and Wiersma D. S, "A Levy flight for light", *Nature*, Vol. 453, pp. 495-498, 2008.

[21] D. Karaboga and B. Basturk, "Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems", *Proceedings of the 12th International Fuzzy Systems Association World Congress, Foundations of Fuzzy Logic and Soft Computing*, pp. 789-798, 2007.

[22] D. Simon, "Biogeography-Based Optimization", *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, pp. 702-713, 2008.

[23] Xin-She Yang and Suash Deb, "Engineering Optimisation by Cuckoo Search", *International Journal of Mathematical Modelling and Numerical Optimization*, Vol. 1, No. 4, pp. 330-343, 2010.

[24] Xin-She Yang and Suash Deb, "Cuckoo search via Lévy flights", *Proceedings of World Congress on Nature & Biologically Inspired Computing*, pp. 210-214, 2009.

[25] Szymon Lukasik and Slawomir Zak, "Firefly algorithm for continuous constrained optimization tasks", *Proceedings of the First International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, pp. 97-100, 2009.

[26] Xin-She Yang, "Firefly algorithm, stochastic test functions and design optimization", *International Journal of Bio-inspired Computation*, Vol. 2, No. 2, pp. 78-84, 2010.

[27] P.W. Tsai, J.S. Pan, B.Y. Liao, M.J. Tsai and V. Istanda, "Bat algorithm inspired algorithm for solving numerical optimization problems", *Applied Mechanics and Materials*, Vol. 148-149, pp. 134-137, 2012.

[28] Xin-She Yang, "Flower pollination algorithm for global optimization", *Proceedings of the 11th International*

Conference on Unconventional Computation and Natural Computation, Vol. 7445, pp. 240-249, 2012.

[29] Amir Hossein Gandomi and Amir Hossein Alavi, "Krill herd: A new bio-inspired optimization algorithm", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 17, No. 12, pp. 4831-4845, 2012.

[30] Taher Niknam, Elahe Taherian Fard, Narges Pourjafarian and Alireza Rousta, "An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering", *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 2, pp. 306-317, 2011.

[31] Yi-Tung Kao, Erwie Zahara and I-Wei Kao, "A hybridized approach to data clustering", *Expert Systems with Applications*, Vol. 34, No. 3, pp. 1754-1762, 2008.

[32] Changsheng Zhang, Dantong Ouyang and Jiaxu Ning, "An artificial bee colony approach for clustering", *Expert Systems with Applications*, Vol. 37, No. 7, pp. 4761-4767, 2010.

[33] Tunchan Cura, "A particle swarm optimization approach to clustering", *Expert Systems with Applications*, Vol. 39, No. 1, pp. 1582-1588, 2012.

[34] Daniela Zaharie, "A Comparative Analysis of Crossover Variants in Differential Evolution", *Proceedings of the International Multiconference on Computer Science and Information Technology*, pp. 171-181, 2007.

[35] Dervis Karaboga and Celal Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", *Applied Soft Computing*, Vol. 11, No. 1, pp. 652-657, 2011.

[36] Yunlong Zhu, Xiaohui Yan, Wenping Zou and Liang Wang, "A new approach for data clustering using hybrid artificial bee colony algorithm", *Neurocomputing*, Vol. 97, pp. 241-250, 2012.

[37] J. Senthilnath, S.N. Omkar and V. Mani, "Clustering using firefly algorithm: performance study", *Swarm and Evolutionary Computation*, Vol. 1, No. 3, pp. 164-171, 2011.

[38] Miao Wan, Lixiang Li, Jinghua Xiao, Cong Wang and Yixian Yang, "Data clustering using bacterial foraging optimization", *Journal of Intelligent Information Systems*, Vol. 38, No. 2, pp. 321-341, 2012.

[39] J. Senthilnath, Vipul Das, S.N. Omkar and V. Mani, "Clustering using Levy Flight Cuckoo Search", *Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications, Advances in Intelligent Systems and Computing*, Vol. 202, pp. 65-75, 2012.

[40] Blake C. L. & Merz C. J, UCI repository of machine learning databases. http://www.ics.uci.edu/-mlearn/MLRepository.html, 1998.

[41] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/