Available online at www.sciencedirect.com

ScienceDirect

Karbala International Journal of Modern Science 4 (2018) 347–360
http://www.journals.elsevier.com/karbala-international-journal-of-modern-science/

Research Article

Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer

Santosh Kumar Majhi*, Shubhra Biswal

Department of Computer Science and Engineering, Veer Surendra Sai Univeristy of Technology, Burla, Odisha, India, 768018

Received 3 May 2018; revised 3 September 2018; accepted 5 September 2018
Available online 24 September 2018

Abstract

K-Means is a popular cluster analysis method which aims to partition a number of data points into K clusters. It has been successfully applied to a number of problems. However, the efficiency of K-Means depends on its initialization of cluster centers. Different swarm intelligence techniques are applied to clustering problem for enhancing the performance. In this work a hybrid clustering approach based on K-means and Ant Lion Optimization has been considered for optimal cluster analysis. Ant Lion Optimization (ALO) is a stochastic global optimization model. The performance of the proposed algorithm is compared against the performance of K-Means, KMeans-PSO, KMeans-FA, DBSCAN and Revised DBSCAN clustering methods based on different performance metrics. Experimentation is performed on eight datasets, for which the statistical analysis is carried out. The obtained results indicate that the hybrid of K-Means and Ant Lion Optimization method performs preferably better than the other three algorithms in terms of sum of intracluster distances and F-measure.

Keywords: Ant Lion Optimization; Clustering; K-Means; Statistical analysis; Stochastic global optimization model

1. Introduction

Clustering is a process of grouping a set of objects based on some similarity measure. Each group of partitioned objects is known as a cluster. The partitioning is performed by clustering algorithms. Hence, clustering is advantageous because it creates the possibility of obtaining previously unknown groups within the same data. Data clustering is an effective method for discovering structure in data sets. Some clustering methods partition objects so that there is no particular boundary among the clusters, whereas some other methods partition objects into mutually exclusive clusters. Also, the distance between two objects is considered as the similarity criteria by some methods.

Clustering algorithms can be categorised into partitioning methods, hierarchical methods, grid based methods and density based methods [1]. Different factors that affect the results of clustering are number of clusters to be formed in a data set, clustering tendency and quality of clustering. Accessing clustering tendency determines whether a non-random structure exists in the data. The existence of a non-random structure in a data set results in meaningful cluster

* Corresponding author.
E-mail addresses: smajhi_cse@vssut.ac.in (S.K. Majhi), shubhrabiswal08@gmail.com (S. Biswal).
Peer review under responsibility of University of Kerbala.

https://doi.org/10.1016/j.kijoms.2018.09.001

analysis. Determining the number of clusters to be formed in a data set is important for few clustering methods in which the number of clusters is used as parameter. To measure the quality of clustering a number of metrics are used. Some methods measure how well the clusters fit the data set, while others measure how well the clusters match the ground truth.

K-Means is one of the well-known partitioned clustering algorithms. Its popularity is due to its simplicity and computational efficiency [1,2]. However, the K-Means algorithm is sensitive to the initial centroids, which is the drawback of the algorithm [3,4,11]. When the initial centroids are changed, then the algorithm gives various solutions. Moreover, the K-Means has a local optima problem. Nowadays, researchers are applying the nature based optimization techniques with clustering algorithms to obtain better clusters and overcome the problems of classical data clustering algorithms. Wang and Lai proposed energy based competitive learning (EBCL) method to handle the significant issues related with competitive learning such as adaptation to clusters having different size and sparsity, auto initialization and outlier problem [5]. Auto-initialization is attained by extracting samples of high energy to form a core point set. They proposed size-sparsity balance of cluster (SSB) and adaptive learning rate based on samples' energy (ALR). SSB method is used for adapting to the clusters of various size and sparsity. ALR is used for eliminating the disturbance created by outliers. Multi-exemplar affinity propagation (MEAP) algorithm is an exemplar based clustering approach proposed by Wang et al. [6]. MEAP addresses the drawback of affinity propagation (AP). AP represents each cluster with the help of single exemplar. Hence, it is unable for modeling the category having multiple subclasses. MEAP model maximizes the sum of the similarities among data points. This approach also maximizes the sum of all linkages between exemplars and super-exemplars. Conscience online learning (COLL) [7] is used to select winning prototype for every datapoint taken randomly for each iteration of the procedure. The winner is updated by the on-line learning rule. Instead of re-computing mean of cluster, the procedure needs only one winning prototype to update. This results in faster convergence of the algorithm. Moreover, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [31] is a popular clustering algorithm which is widely used in many applications. Some of the advanced variance of the DBSCAN is also used for specific applications. One of the variant is revised DBSCAN [32]. It solves the problem of detecting border objects of adjacent clusters. Das et al. [33] proposed k-mer in composition vector method for comparing genome sequences.

To optimize the objective functions of clustering algorithm, different evolution strategies and population based optimization strategies are applied. Objective function can be centroid or non-centroid type of functions. Evolution strategies are used to optimize both centroid and non-centroid objective functions. Evolution strategies are applied parallel to the clustering problems. The approach is applied on different data sets to display the usefulness of evolution strategies. A hybrid genetic algorithm is proposed to find optimal partition of data into K-clusters [8]. Kader [9] proposed a hybrid of genetic algorithm and K-Means clustering and verified the efficiency of the algorithm by applying in a practical scenario on online shopping market segmentation case. Hassanzadeh and Meybodi have proposed a firefly optimization based clustering algorithm and compared the obtained results with PSO, K-Means, and K-PSO considering standard datasets [10]. Han et al. [11] proposed a clustering algorithm based on the Bird Flock Gravitational Search Algorithm (BFGSA). The algorithm is compared with the GSA, the Artificial Bee Colony (ABC), the Firefly Algorithm (FA), K-Means and different variants of Particle Swam Optimization such as NM-PSO, K-PSO, K-NM-PSO, and CPSO. The experimental results show better performance of BFGSA over other compared algorithms. Firouzi et al. introduced a hybrid Simulated annealing and ant colony optimization based data clustering algorithm [12]. Kao et al. has implemented the hybrid of PSO and K-Means [13] and compared with Genetic Algorithm based clustering [14] and hybrid of GA and K-Means [15] to improve the clustering result. This algorithm has a better convergence characteristic with a few numbers of evaluations. However, the main drawback is having the overlapped data points. A new data clustering approach is proposed in paper [16] based on PSO integrated with the kernel density estimation (KDE). KDE is used to improve the balance between exploitation and exploration. The hybridization of improved PSO and genetic algorithm (GA) along with K-Means algorithm improves the convergence speed as well as helps to find the global optimal solution. In the first stage, IPSO has been used to get a global solution in order to get optimal cluster canters. Then, the crossover steps of GA are used to improve the quality of particles and mutation is used for diversification of solution space in order to avoid premature convergence. In paper [17], a hybrid K-Means based on improved PSO and GA has been proposed which is used to find the global optimal solution and also results in improved

convergence speed. Improved PSO is used to find the optimal cluster centres by finding the global optimal solution. The quality of clustering is improved by the crossover steps of GA and mutation is applied to avoid premature convergence.

It is clear from the literature that with evolution of new optimization techniques, researchers are using the techniques for developing data clustering algorithms. Data clustering algorithms using GA (genetic algorithm) is based on the selection, crossover and mutation. Whereas Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Firefly Optimization (FO), Ant Colony Optimization (ACO), Simulated Annealing (SA), Gravitational Search Algorithm (GSA), Bird Flock Gravitational Search Algorithm (BFGSA), Artificial Bee Colony (ABC) are inspired from the natural phenomena and natural organisms and used to improve the performance of clustering. Furthermore, these meta-heuristic optimization methods search for global optimal solutions for the functions using a collaborative approach of search agents and significant parameters within the search region. Consequently, the optimization algorithm has two important performance indexes in terms of exploration and exploitation. Moreover, randomness plays a major role in optimization methods to generate distinct result in each run; however it is still difficult to avoid local minima problem. Swarm and nature based optimizations like PSO, DE, GA etc. have the problem of early convergence. The problem of early convergence has been solved by using some hybridization techniques and innovative variants of the optimization methods.

As reported in literature, a number of advanced meta-heuristic algorithms like GWO, ABC etc uses distinct functions in the process of exploitation and exploration. Moreover, according to Wolpert and Macready [18], none of the techniques will solve all the optimization problems. Furthermore, the no free lunch theorem says that the performance of an algorithm does not assure its success in various sets of problems. This motivates us to apply Antlion Optimizer (ALO) algorithm for data clustering problem. Antlion Optimizer (ALO) algorithm is a novel meta-heuristic algorithm inspired by the behaviour of antlion proposed by Mirjalil [19]. It maintains the balance between exploration and exploitation by using a global search for exploration and local search for exploitation. The ALO method has not only high potential to explore the search space but also has the high exploitation capability to quickly converge for a global optimum. The ALO algorithm has been considered due to its faster convergence, effective exploration using random walks and random selection of search agents and accurate exploitation using adjustive limits of traps. Nowadays, this algorithm is used in many engineering domains like automobile cruise control system [20,21], power system [22,23], classification and neural network [24,25] etc.

By considering the discussed advantages of the ALO, in this work, a data clustering algorithm is proposed by combining the K-Means with ALO. The performance of the proposed KMeans-ALO clustering algorithm is compared against the performance of K-Means, KMeans-PSO, KMeans-FA, DBSCANand Revised DBSCAN. KMeans-PSO combines K-Means clustering algorithm with Particle Swarm Optimization (PSO) for optimization of the cluster centroids. Similarly Firefly Algorithm (FA) is combined with KMeans in KMeans-FA hybrid clustering method.

The rest of the paper is organised as follows. Section 2 describes the materials and methods required which gives the description about the algorithms used, data sets and the performance metrics based on which the quality of clustering is measured. Section 3 represents the results obtained from the implementation of the algorithms followed by conclusion of the work in the last section.

2. Materials and methods

2.1. Proposed hybrid K-Means and ALO clustering algorithm

K-Means clustering is an unsupervised hard partitioning clustering method. The objective is to find k clusters from the data based on the objective function J given in Eq. (1).

$$J = \sum_{i=1}^{k} \sum_{j=1}^{N} d^2\left(C_i - X_j\right) \tag{1}$$

where $d^2(C_i\text{-}X_j)$ is the squared Euclidean distance between ith cluster centroid and jth data point. N is the total number of data points. Based on the distance obtained, the points are assigned to the cluster with minimum distance from the centroid. After the points are clustered the mean of all points belonging to the cluster is found. Then mean value is assigned as the new cluster centroid for the next iteration. This process is repeated until the centroid obtained is same as that of the previous iteration. The aim of K-Means algorithm is to minimize the objective function.

Ant Lion Optimization method is also a nature inspired algorithm which follows the hunting behaviour of antlion larvae [19]. An ant lion larva creates a conical shaped hole by moving along a circular path in the sand

and throwing the sand with its huge jaw. After digging the trap, larvae hide at the bottom of the cone and waits for the ants to be trapped in the pit. Once the ant lion realizes that a prey has been caught in the trap, the ant lion throws sand outwards and slips its prey into the pit. When a prey is caught into the jaw, the ant lion pulls the prey toward itself and consumes. This process is mathematically designed to perform optimization. There are five main steps in this method are: (1) Random walks of ants, (2) Building traps, (3) Entrapment of ants in traps, (4) Catching preys and (5) Rebuilding trap.

Ants use random walks for moving around the search space which is affected by the traps of antlions. The positions of ants are updated with random walk at every iteration. The random walks for iteration t are created using. (2). However to ensure that all the random walks fall inside the boundary of search space, normalization is applied. The random walks are normalized using Eq. (3).

$$X[t] = [0, \text{cumsum}(2r(t_1) - 1), \text{cumsum}(2r(t_2) - 1), \ldots \text{cumsum}(2r(t_n) - 1)] \quad (2)$$

where, $r(t) = 1$ if $rand > 0.5$ or 0 if $rand \leq 0.5$

$$X_i^t = \frac{(X_i^t - a_i) \times (b_i - c_i^t)}{(d_i^t - a_i)} + c_i. \quad (3)$$

a_i and b_i are the minimum and maximum of random walk respectively for i^{th} variable. c_i^t is the minimum of i^{th} variable at iteration t. d_i^t is the maximum of i^{th} variable at iteration t.

The traps created by antlions impact the random walks of ants. This process is mathematically explained using Eq. (4) and Eq. (5).

$$c_i^t = Antlion_j^t + c^t \quad (4)$$

$$d_i^t = Antlion_j^t + d^t \quad (5)$$

here $Antlion_j^t$ is the position of antlion i at iteration t. c^t and d^t are the minimum and maximum of all variables respectively. c_i^t is the minimum of i^{th} ant at iteration t and d_i^t is the maximum of i^{th} ant at iteration t.

A roulette wheel selection approach is used to select antlions for optimization based on their fitness value. The fittest antlion obtained in each iteration is saved as elite. The elite impact the movements of ants. Furthermore the positions of ants are updated based on the random walk of selected antlion as well as the elite because every ant walks around a selected antlion and also around the elite. This process is formulated in eq. (6).

$$Ant_i^t = \frac{R_A^t + R_E^t}{2}. \quad (6)$$

R_A^t is the random walk around the antlion selected by the roulette wheel at t^{th} iteration and R_E^t is the random walk around the elite at t^{th} iteration. Ant_i^t is the position of ant i at iteration t.

The fitness values of all ants are calculated. An antlion is replaced by corresponding ant if the ant has better fitness than the antlion. Similarly the elite is also replaced by an antlion if the antlion has better fitness than elite.

For the improvement of the quality of clustering of KMeans algorithm a hybrid of KMeans clustering method and antlion optimization algorithm is proposed. In the first step the number of clusters to be formed is determined. Then all the data points are clustered based on minimum Euclidean distance obtained. The next step is to calculate the optimized cluster centroid for each of the clusters obtained. For the optimization process each cluster is initialized as ant and antlion population randomly. Then the fitness value of all the ants and antlions are calculated using the objective function of the KMeans clustering method. As the sum of average of intracluster distances should be minimized, the antlion having the minimum fitness value is considered as elite. For each cluster the Antlion optimization process is carried out to obtain the best position of the cluster centroid. The returned elites are treated as the centroids for the K-Means clustering algorithm. The flow chart of the proposed method is given in Fig. 1.

Hybrid K-Means and ALO clustering algorithm

INPUT:The number of iterations*T*, number of clusters *K*, number of ants *A*, number of antlions *L*, total ant antlion population *P*, dataset D with *N*instances and *M* attributes.

OUTPUT: Optimized cluster centroids.

BEGIN ALGORITHM:	No. of Operations
Select K random points as cluster centres	
WHILE the end criterion is not satisfied	*(T+1)*
FOR each point	*T*(P+1)*
Find the Euclidean distance of each point from the cluster centres	*T*P*K*
Assign the point to the cluster with minimum Euclidean distance	*T*K*
END FOR	
Compute the mean of all points in each cluster	*T*K*
Assign the mean values as the new cluster centres	*T*K*
END WHILE	
Return K clusters	
FOR each cluster	*(K+1)*
Initialize the first population of ants from the dataset	

(*continued on next page*)

(*continued*)

BEGIN ALGORITHM:	No. of Operations
Initialize the first population of ant lions randomly	
Calculate the fitness of ants and ant lions using the objective function	*K*P*
Select the ant lion with minimum fitness value as elite	*K*L*
WHILE the end criterion is not satisfied	*K*(T+1)*
FOR every ant	*T*K*(A+1)*
Select an ant lion using Roulette wheel	*T*K*A*
Update minimum of all variables and maximum of all variables	*T*K*A*
Create a random walk as using Eq. (2)	*T*K*A*
Normalize the random walk using Eq. (3)	*T*K*A*
Update the position of ant by Eq. (6)	*T*K*A*
END FOR	
Find the fitness value of all ants	*T*K*
Replace an antlion with its corresponding ant if *f(Anti)<f (Antlionj)*	*T*K*
Update elite if *f (Antlionj) <f (elite)*	*T*K*
END WHILE	
Return elite	*K*
END FOR	
Select elite as the new center of the cluster	*1*
END ALGORITHM	

2.2. Data sets

The algorithms mentioned in the previous section are applied on 8 different datasets to obtain the results. The datasets are collected from UCI machine learning repository [26]. The data sets are glass, vowel, ionosphere, leaf, gene expression cancer RNA-seq, waveform database generator (version 2), immunotherapy and soybean. All the instances of Glass dataset are divided into 7 classes which defines the types of glasses. Vowel dataset is the connectionist bench (vowel recognition − deterding data) data set. The ionosphere data set can be categorized into 2 classes based on the radar returns. This checks for free electrons in the ionosphere. The instances of the Leaf dataset data are collected from 40 different plant species. The Gene expression cancer RNA-seq dataset is a collection of randomly extracted gene expressions of patients having different types of tumor such as: BRCA, KIRC, COAD, LUAD, and PRAD. The waveform data set consists of 5000 data instances which are categorized into 3 classes of waves. The immunotherapy data set is a collection of treatment results of patients using immunotherapy. The classes formed based on the result of the treatment (whether positive or negative). The soybean data set is a collection of 4 classes of soybean i.e. D1, D2, D3, D4. A brief description about all the datasets discussed above is given in Table 1.

2.3. Performance metrics

The quality of clustering is obtained using the proposed algorithm is evaluated based on different performance metrics. The performance metrics considered for this work are average of sum of intra-cluster distances and F-measure.

2.3.1. Average of sum of intracluster distances

Data points belonging to same cluster should be as close as possible i.e. the intracluster distance should be minimum in order to get optimal quality of clustering. Different methods to evaluate the intracluster distance are complete diameter, centroid diameter and average diameter. In this work the centroid diameter method of calculating the intracluster distance is considered.

Complete diameter method calculates the intra-cluster distance by calculating the greatest distance between any two points belonging to the cluster. In centroid diameter method the intracluster distance is the average distance between the cluster centroid and all points belonging to that cluster. In this work intra-cluster distance is being measured using the centroid method of finding the intra-cluster distance. Average diameter method considers the average distance between all pairs of points belonging to the cluster.

2.3.2. F-measure

F-measure is calculated using the concepts of precision and recall from information retrieval. Each class i of the data set is regarded as the set of n_i items desired for a query. Each cluster j is regarded as the set ofn_j items retrieved for a query. n_{ij} represents the number of elements of class i within cluster j. For each class i and cluster j, precision and recall calculations is represented in Eq. (7) and F-measure is given in Eq. (8). The final calculation of the F-measure is given in Eq. (9). Here b is a constant which is responsible for equal weighing for precision and recall. The value of b is taken as 1.

$$precision(i,j) = \frac{n_{ij}}{n_j},\ recall(i,j) = \frac{n_{ij}}{n_i} \tag{7}$$

$$F(i,j) = \frac{(b^2+1).precision(i,j).recall(i,j)}{b^2.precision(i,j) + recall(i,j)} \tag{8}$$

$$F = \sum_{i=1}^{k} \frac{n_i}{N} max\{F(i,j)\} \tag{9}$$

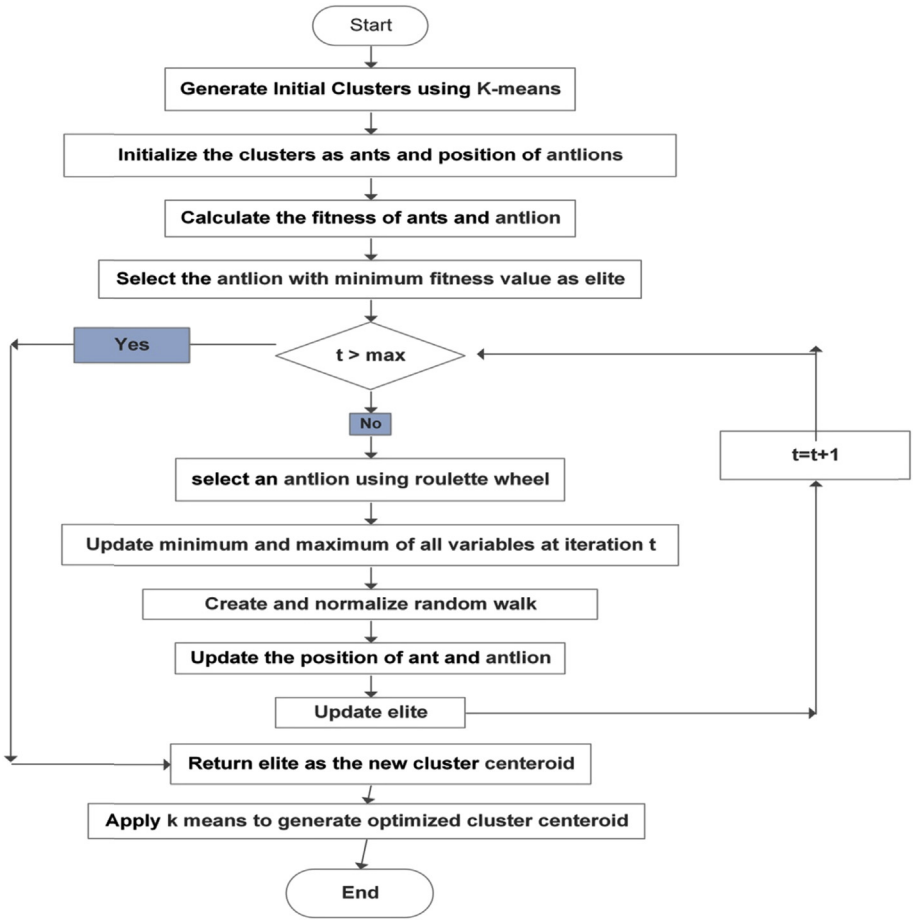

Fig. 1. Flow chart of KMeans-ALO.

3. Results and discussion

All the algorithms discussed in the materials and methods section are implemented with MatlabR2016a on a Windows platform using Intel(R) Core(TM) i3-2310 M, 2.10 GHz, 4 GB RAM computer. The experimental results for the average of sum of intracluster distance are calculated on all the eight datasets discussed in the materials & methods section are provided in Tables 2–9. The results are collected over 10 different runs, for 100, 500 and 1000 iterations.

Table 1
A brief description of the data sets used.

Sl. No.	Dataset	No. of instances	No. of classes	No. of Attributes
1	Glass	214	7	10
2	Vowel	990	6	10
3	Ionosphere	351	2	34
4	Leaf	340	30	16
5	RNA-seq	801	5	20,531
6	Waveform	5000	3	40
7	Immunotherapy	90	2	8
8	Soybean	47	2	35

For glass dataset all the three hybrid methods i.e. KMeans-ALO, KMeans-FA and KMeans-PSO gives the same minimum value of 0.2663 for sum of average of intracluster distances for 100 iterations. For 500 iterations both KMeans-PSO and KMeans-FA give minimum intracluster distance i.e. 0.2663. For 1000 iterations KMeans-PSO and KMeans give minimum intracluster distance. The results given in Table 2 are obtained over 214 data points belonging to two different attributes. For vowel dataset the sum of average of intracluster distance, F-measure and standard deviation have been evaluated over 528 data points. The minimum intracluster distance value of 1.5739 and 1.5744 is obtained by KMeans-ALO for

Table 2
Results obtained by K-Means, KMeans-PSO, KMeans-FA, DBSCAN, Revised DBSCAN and KMeans-ALOalgorithms for 10 different runs on Glass data set for 100, 500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	0.2696	0.27,678	0.3053	0.9989	0.0174
KMeans-PSO	100	0.2663	0.28,374	0.3055	0.9989	0.0186
KMeans-FA	100	0.2663	0.28,758	0.3051	0.9989	0.0184
DBSCAN	100	0.2681	0.2681	0.2681	0.0280	0
Revised DBSCAN	100	0.2669	0.2669	0.2669	0.8928	0
KMeans-ALO	100	0.2663	0.28,287	0.3051	0.9989	0.0192
K-Means	500	0.2665	0.28,674	0.3053	1.4320	0.0195
KMeans-PSO	500	0.2663	0.291	0.3053	0.9989	0.0184
KMeans-FA	500	0.2663	0.28,078	0.3051	1.4320	0.0169
DBSCAN	500	0.2681	0.2681	0.2681	0.0280	0
Revised DBSCAN	500	0.2669	0.2669	0.2669	0.8928	0
KMeans-ALO	500	0.2664	0.29,027	0.3051	0.9989	0.0192
K-Means	1000	0.2665	0.2865	0.3057	1.4320	0.0199
KMeans-PSO	1000	0.2664	0.27,998	0.3051	0.9989	0.0173
KMeans-FA	1000	0.2697	0.29,165	0.3051	0.9989	0.0173
DBSCAN	1000	0.2681	0.2681	0.2681	0.0280	0
Revised DBSCAN	1000	0.2669	0.2669	0.2669	0.8928	0
K-Means-ALO	1000	0.2664	0.29,027	0.3051	0.9989	0.0192

Table 3
Results obtained by K-Means, KMeans-PSO, KMeans-FA, DBSCAN, Revised DBSCAN and KMeans-ALO algorithms for 10 different runs on Vowel data set for 100,500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	8.1460	10.31,046	14.4252	1.3595	2.1627
KMeans-PSO	100	1.5743	1.66,803	2.0370	1.7035	0.3292
KMeans-FA	100	1.6617	1.76,214	2.0222	1.7035	0.1527
DBSCAN	100	1.9747	1.9747	1.9747	0.0061	0
Revised DBSCAN	100	1.8908	1.8908	1.8908	0.9671	0
KMeans-ALO	100	1.6601	1.67,515	2.0230	1.6901	0.1539
K-Means	500	8.1437	9.76,398	14.4389	1.3451	2.6978
KMeans-PSO	500	1.5856	1.5703	2.0314	1.7533	0.3138
KMeans-FA	500	1.6626	1.78,148	2.1583	1.7035	0.1675
DBSCAN	500	1.9747	1.9747	1.9747	0.0061	0
Revised DBSCAN	500	1.8908	1.8908	1.8908	0.9671	0
KMeans-ALO	500	1.5739	1.62,269	2.0233	1.7535	0.1885
K-Means	1000	8.1485	9.41,156	14.4443	1.3945	2.6497
KMeans-PSO	1000	1.5750	1.69,249	2.0238	1.7147	0.2015
KMeans-FA	1000	1.6507	1.68,894	1.8941	1.7035	0.0725
DBSCAN	1000	1.9747	1.9747	1.9747	0.0061	0
Revised DBSCAN	1000	1.8908	1.8908	1.8908	0.9671	0
KMeans-ALO	1000	1.5744	1.64,987	2.0265	1.7201	0.2380

500 and 1000 iterations respectively. For 100 iterations Kmeans-PSO gives minimum intracluster distance. For 500 and 1000 iterations KMeans-ALO gives maximum F-measure of 1.7201 and 1.7535 respectively. For 100 iterations KMeans-PSO and KMeans-FA give maximum F-measure value of 1.7035. In Table 4 the sum of average of intracluster distances, F-measure and standard deviation for all the four algorithms are tabulated for ionosphere data set. The values are evaluated over 351 data points of ionosphere data set. KMeans-ALO provides minimum intracluster distance of 0.6659. For leaf data set, DBSCAN provides minimum intracluster distance of 0.0512. However the maximum value for F-measure is obtained by kmeans-ALO. The results are evaluated on 340 data points of two different attributes. For gene expression cancer RNA-seq data set DBSCAN gives the minimum intracluster distance value i.e. 3.1722. Maximum F-measure value of 1.3445 is obtained by KMeans-FA. The results given in Table 6 are obtained over 801

Table 4
Results obtained by K-Means, KMeans-PSO, KMeans-FA, Revised DBSCAN and KMeans-ALO algorithms for 10 different runs on Ionosphere data set for 100,500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	0.6666	0.8759	1.5381	20.1548	0.2740
KMeans-PSO	100	0.6663	0.78,042	1.5380	19.6610	0.2694
KMeans-FA	100	0.6664	0.70,577	0.8608	19.6600	0.0817
DBSCAN	100	1.5490	1.5490	1.5490	17.1000	0
Revised DBSCAN	100	1.0121	1.0121	1.0121	19.0542	0
KMeans-ALO	100	0.6663	0.72,899	1.1135	20.1548	0.1398
K-Means	500	0.6669	0.85,855	1.5381	34.6548	0.3622
KMeans-PSO	500	0.6664	0.84,111	1.5379	19.6610	0.3672
KMeans-FA	500	0.6660	0.75,412	1.5379	19.7866	0.2753
DBSCAN	500	1.5490	1.5490	1.5490	17.1000	0
Revised DBSCAN	500	1.0121	1.0121	1.0121	19.0542	0
KMeans-ALO	500	0.6660	1.11,551	1.5381	20.1548	0.3447
K-Means	1000	0.7539	0.88,422	1.6681	34.3522	0.0273
KMeans-PSO	1000	0.6663	0.85,031	1.5379	34.9099	0.3635
KMeans-FA	1000	0.6672	0.80,164	1.1124	19.9295	0.2149
DBSCAN	1000	1.5490	1.5490	1.5490	17.1000	0
Revised DBSCAN	1000	1.0121	1.0121	1.0121	19.0542	0
KMeans-ALO	1000	0.6659	0.84,095	1.5382	29.1508	0.3675

Table 5
Results obtained by K-Means, KMeans-PSO, KMeans-FA, Revised DBSCAN and KMeans-ALO algorithms for 10 different runs on Leaf data set for 100,500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	0.0584	0.05856	0.0587	2.9701	0.0002
KMeans-PSO	100	0.0583	0.0584	0.0587	2.9777	0.0001
KMeans-FA	100	0.0582	0.05855	0.0589	2.9751	0.0002
DBSCAN	100	0.0512	0.0512	0.0512	2.9708	0
Revised DBSCAN	100	0.0550	0.0550	0.0550	2.9777	0
KMeans-ALO	100	0.0583	0.0583	0.0583	2.9787	0
K-Means	500	0.0584	0.05852	0.0586	2.9701	0.0002
KMeans-PSO	500	0.0583	0.0585	0.0587	2.9777	0.0002
KMeans-FA	500	0.0582	0.05838	0.0586	2.9751	0.0001
DBSCAN	500	0.0512	0.0512	0.0512	2.9708	0
Revised DBSCAN	500	0.0550	0.0550	0.0550	2.9777	0
KMeans-ALO	500	0.0583	0.05834	0.0585	2.9787	0.00008
K-Means	1000	0.0584	0.06414	0.0586	2.9701	0.0002
KMeans-PSO	1000	0.0583	0.0584	0.0587	2.9777	0.0002
KMeans-FA	1000	0.0582	0.05855	0.0588	2.9751	0.0002
DBSCAN	1000	0.0512	0.0512	0.0512	2.9708	0
Revised DBSCAN	1000	0.0550	0.0550	0.0550	2.9777	0
KMeans-ALO	1000	0.0583	0.0583	0.0583	2.9787	0

data points belonging to two different attributes. For waveform database generator data set we have considered 5000 data points belonging to 2 attributes and evaluated the sum of average of intracluster distance for 100, 500 and 1000 iterations on 10 different runs. Also the F-measure and standard deviation is calculated for 100, 500 and 1000 iterations for each of the four algorithms used. It can be observed from Table 7 that for this dataset the KMeans-ALO gives the minimum intracluster distance of 3.5675 whereas maximum value for F-measure is obtained by DBSCAN. For immunotherapy dataset KMeans-ALO gives the minimum intracluster distance value i.e. 274.1204 and maximum F-measure value of 1.6276. The results given in Table 8 are obtained over 90 data points of two different attributes. For soybean data set the sum of average of intracluster distance, F-measure and standard deviation are obtained over 47 data points

Table 6
Results obtained by K-Means, KMeans-PSO, KMeans-FA, Revised and KMeans-ALO algorithms for 10 different runs on Gene expression cancer RNA-seqdata set for 100, 500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	3.3630	3.3538	3.3757	1.1818	0.0063
KMeans-PSO	100	3.3591	3.36,451	3.3687	1.1818	0.0034
KMeans-FA	100	3.3636	3.36,823	3.3708	1.3445	0.0031
DBSCAN	100	3.1722	3.1722	3.1722	1.3440	0
Revised DBSCAN	100	3.3698	3.3698	3.3698	1.3239	0
KMeans-ALO	100	3.3534	3.35,598	3.3579	1.1818	0.0016
K-Means	500	3.3620	3.5694	3.38,601	1.1818	0.0065
KMeans-PSO	500	3.3636	3.44,384	4.1410	1.1818	0.2324
KMeans-FA	500	3.3562	3.35,928	3.3606	1.3445	0.0021
DBSCAN	500	3.1722	3.1722	3.1722	1.3440	0
Revised DBSCAN	500	3.3698	3.3698	3.3698	1.3239	0
KMeans-ALO	500	3.3534	3.35,629	3.3579	1.3442	0.0597
K-Means	1000	3.3561	3.3746	3.36,674	1.1818	0.0042
KMeans-PSO	1000	3.3590	3.36,425	3.3689	1.1818	0.0030
KMeans-FA	1000	3.3557	3.37,016	3.3757	1.3445	0.0079
DBSCAN	1000	3.1722	3.1722	3.1722	1.3440	0
Revised DBSCAN	1000	3.3698	3.3698	3.3698	1.3239	0
KMeans-ALO	1000	3.3536	3.35,629	3.3579	1.3442	0.0598

Table 7
Results obtained by K-Means, KMeans-PSO, KMeans-FA, Revised DBSCAN and KMeans-ALO algorithms for 10 different runs on Waveform database generator data set for 100, 500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	3.5678	3.79,873	4.1462	0.9893	0.2321
KMeans-PSO	100	3.5768	3.67,325	4.1669	1.0880	0.1703
KMeans-FA	100	3.5684	3.74,025	4.1411	0.9914	0.2766
	100	3.5708	3.5708	3.5708	1.1031	0
Revised DBSCAN	100	3.5690	3.5690	3.5690	1.0999	0
KMeans-ALO	100	3.5675	4.44,033	4.1409	1.0014	0.2292
K-Means	500	3.5692	4.32,885	4.1577	0.9893	0.2818
KMeans-PSO	500	3.5687	3.85,696	4.1417	1.0880	0.7208
KMeans-FA	500	3.5685	3.79,754	4.1411	1.0880	0.2956
DBSCAN	500	3.5708	3.5708	3.5708	1.1031	0
Revised DBSCAN	500	3.5690	3.5690	3.5690	1.0999	0
KMeans-ALO	500	3.5682	3.8798	4.1530	1.0013	0.2743
K-Means	1000	3.5682	3.67,452	4.1408	0.9914	0.2825
KMeans-PSO	1000	3.7028	3.73,201	4.1520	1.0880	0.2888
KMeans-FA	1000	3.5684	3.683	4.1411	0.9914	0.2414
DBSCAN	1000	3.5708	3.5708	3.5708	1.1031	0
Revised DBSCAN	1000	3.5690	3.5690	3.5690	1.0999	0
KMeans-ALO	1000	3.5675	4.44,033	4.1409	1.0014	0.2743

of two different attributes. KMeans-ALO gives the minimum intracluster distance value i.e. 2.0198 for both 500 and 1000 iterations. For 100 iterations KMeans-PSO gives the minimum value for sum of average of sum of intracluster distance.

The values in Table 2 to Table 9 show that KMeans-ALO gives minimum intracluster distance and maximum F-measure for glass, vowel, ionosphere, waveform database generator (version 2), immunotherapy and soybean. For leaf dataset Kmeans-FA provides minimum intracluster distance and for gene expression cancer RNA-seq DBSCAN provides minimum intracluster distance. The KMeans-ALO uses random walks and random selection of search agents for effective exploration and uses adjustive limits of traps for accurate exploitation. Thus KMeans-ALO provides better results as compared to other methods considered by quickly converging for global optimum.

Table 8
Results obtained by K-Means, KMeans-PSO, KMeans-FA, Revised DBSCAN and KMeans-ALO algorithms for 10 different runs on Immunotherapy data set for 100, 500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	274.9177	342.66,974	625.3499	1.6040	131.8395
KMeans-PSO	100	283.5370	357.65,789	603.9673	1.6073	162.2166
KMeans-FA	100	274.7445	345.97,009	626.0714	1.6273	147.6856
DBSCAN	100	298.3393	298.3393	298.3393	0.0167	0
Revised DBSCAN	100	292.0989	292.0989	292.0989	0.9124	0
KMeans-ALO	100	274.6798	325.54,186	591.5543	1.6276	131.1642
K-Means	500	283.9177	274.69,255	276.1581	1.6040	87.1694
KMeans-PSO	500	283.5370	324.67,073	603.9673	1.6073	140.9375
KMeans-FA	500	276.1116	276.12,675	276.1621	1.6276	0.02439
DBSCAN	500	298.3393	298.3393	298.3393	0.0167	0
Revised DBSCAN	500	292.0989	292.0989	292.0989	0.9124	0
KMeans-ALO	500	274.1204	339.94,343	603.0107	1.6276	131.5336
K-Means	1000	274.4723	310.43,787	276.4448	1.6028	95.1210
KMeans-PSO	1000	283.5370	299.79,307	603.9673	1.6076	102.7227
KMeans-FA	1000	274.6302	411.61,469	626.2808	1.6273	175.3127
DBSCAN	1000	298.3393	298.3393	298.3393	0.0167	0
Revised DBSCAN	1000	292.0989	292.0989	292.0989	0.9124	0
KMeans-ALO	1000	274.1204	328.6323	624.6784	1.6276	140.9650

Table 9
Results obtained by K-Means, KMeans-PSO, KMeans-FA, Revised DBSCAN and KMeans-ALO algorithms for 10 different runs on Soybean data set for 100, 500 and 1000 iterations.

Methods	Iterations	Best value	Average value	Worst Value	F-measure	Standard deviation
K-Means	100	2.0215	2.9984	3.0859	6.7556	0.1809
KMeans-PSO	100	2.0201	2.49,647	3.0861	6.7556	0.4827
KMeans-FA	100	2.1663	2.60,784	3.0852	7.5887	0.5027
DBSCAN	100	2.1963	2.1963	2.1963	4.2600	0
Revised DBSCAN	100	2.1921	2.1921	2.1921	4.9801	0
KMeans-ALO	100	2.1640	2.51,855	3.0823	7.5887	0.4622
K-Means	500	2.0215	2.39,974	3.0850	7.7195	0.4576
KMeans-PSO	500	2.0199	2.34,042	3.0852	7.6312	0.3817
KMeans-FA	500	2.0319	2.67,646	3.0859	7.6312	0.5272
DBSCAN	500	2.1963	2.1963	2.1963	4.2600	0
Revised DBSCAN	500	2.1921	2.1921	2.1921	4.9801	0
KMeans-ALO	500	2.0198	2.53,986	3.0830	7.3526	0.4619
K-Means	1000	2.5239	2.39,894	3.0848	7.3526	0.4369
KMeans-PSO	1000	2.0199	2.67,777	3.0853	7.3526	0.5317
KMeans-FA	1000	2.0218	2.70,866	3.0848	6.7556	0.4474
DBSCAN	1000	2.1963	2.1963	2.1963	4.2600	0
Revised DBSCAN	1000	2.1921	2.1921	2.1921	4.9801	0
KMeans-ALO	1000	2.0198	2.5326	3.0830	7.3526	0.4516

3.1. Asymptotic analysis of the proposed KMeans-ALO clustering method

Asymptotic analysis of an algorithm is the run time performance of the algorithm which depends on the input to the algorithm. An algorithm will work in a constant time if it does not depend on the input. The effective time complexity of K-Means algorithm is known to be $O\ (n^2)$ [27]. Here, we evaluated the performance of the algorithm in terms of input size. The input parameters on which the proposed KMeans-ALO depends are number of iterations (T), number of clusters to be formed (K), total population of ants and antlions (P), number of ants (A), number of antlions (L), number of attributes (M) and number of instances of the dataset. The multiplication of the steps needed for the algorithm and the cost involved in each step gives the total cost of execution for the proposed

Table 10
Average ranking of the clustering algorithms based on average of sum of intracluster distances.

Data set	K-Means	KMeans-PSO	KMeans-FA	DBSCAN	Revised DBSCAN	KMeans-ALO
Glass	0.2665 (3)	0.2664 (1.5)	0.2697 (6)	0.2681 (5)	0.2669 (4)	0.2664 (1.5)
Vowel	8.1485 (6)	1.5750 (2)	1.6507 (3)	1.9747 (5)	1.8908 (4)	1.5744 (1)
Ionosphere	0.7539 (4)	0.6663 (2)	0.6672 (3)	1.5490 (6)	1.0121 (5)	0.6659 (1)
Leaf	0.05834 (6)	0.0583 (4.5)	0.0582 (3)	0.0512 (1)	0.0550 (2)	0.0583 (4.5)
RNA-seq	3.3561 (4)	3.3590 (5)	3.3557 (3)	3.1722 (1)	3.3698 (6)	3.3536 (2)
Waveform	3.5682 (2)	3.7028 (6)	3.5684 (3)	3.5708 (4)	3.5690 (5)	3.5675 (1)
Immunotherapy	274.4723 (2)	283.5370 (4)	274.6320 (3)	298.3393 (6)	292.0989 (5)	274.1204 (1)
Soybean	2.5239 (6)	2.0199 (2)	2.0218 (3)	2.1963 (5)	2.1921 (4)	2.0198 (1)
Average rank(Rj)	**4.125**	**3.375**	**3.375**	**4.125**	**4.375**	**1.625**

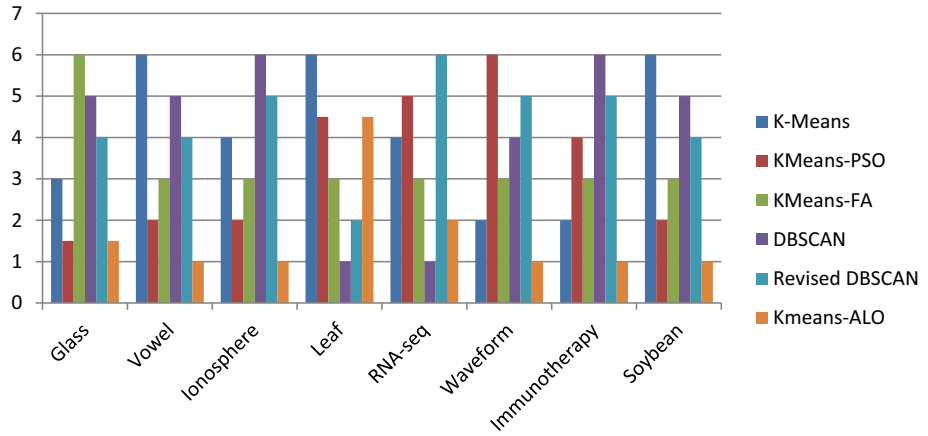

Fig. 2. Comparison of intracluster distances obtained using K-Means, KMeans-PSO, KMeans-FA, DBSCAN, Revised DBSCAN and KMeans-ALO for all the 8 datasets based on the average rank of the algorithms.

Table 11
Results obtained from Holm's procedure.

i	Algorithms	*z*-value	*p*-value	*α/(k-i)*	Hypothesis
1	Kmeans	−2.8571	0.00219	0.02	Rejected
2	Kmeans-PSO	−2	0.02275	0.025	Rejected
3	Kmeans-FA	−2	0.02275	0.0333	Rejected
4	DBSCAN	−2.8571	0.00219	0.05	Rejected
5	Revised DBSCAN	−3.1428	0.00084	0.10	Rejected

algorithm. The cost associated with each step is assumed to be 1 unit. The total number of operations for KMeans-ALO is determined from the algorithm given in Section 2.1.

To access the input dataset having N instances and M attributes ($N*M$) number of additional operations are needed.

The total cost is represented as a function of T, K, P, A, L, N and M, which is given as

$$f(T,K,P,A,L,N,M) = TPKNM + 6TKA + 8TK + TP + KL + KPNM + 3K + 2T + 3$$

To obtain the worst case performance of the proposed Kmeans-ALO it is assumed that all the parameters are equal, resulting in Eq. (10).

$$\begin{aligned}\text{Total number of operations} &= (T+1) + [T*(P+1)] + TPK + TK + TK + TK + (K+1) + KP + [K*(T+1)] \\ &\quad + [T*K*(A+1)] + TKA + KL + TKA + TKA + TKA + TKA + TK + TK + TK \\ &\quad + K + 1 \\ &= TPK + 6TKA + 8TK + TP + KL + KP + 3K + 2T + 3\end{aligned}$$

$$f(n)= n^5+n^4+6n^3+10n^2+5n+3 \quad (10)$$

From Eq. (7), it can be observed that $f(n) = O(n^5)$ for $n \geq 1$. It specifies that the algorithm runs in polynomial time and the time complexity of the algorithm in extremely worst case is $O(n^5)$.

The time complexity of K-Means is $O(n^2)$ and the time complexity of KMeans-ALO is $O(n^5)$. However, both K-Means and KMeans-ALO are solvable in polynomial time.

3.2. Statistical performance evaluation

To determine the existence of significant differences among the performance of the clustering algorithms statistical analysis is performed. For determining the differences Friedman test has been employed in this work. Friedman test is a non-parametric test which is used to find differences among groups for ordinal dependent variables [28]. The null hypothesis $H_0$is considered as

H_0: All the four clustering algorithms perform equally.

The level of confidence α for the test is taken as 0.05. Ranks are assigned to each algorithm based on their predictive accuracy, ranged from 1 to k. In this work, as we are considering intracluster distance as one of the performance metrics, the algorithms are ranked based on the intracluster distance obtained by them. The algorithm which results in minimum value of average of sum of intracluster distance gets the rank 1 and the algorithm with maximum value of average of sum of intracluster distance gets the maximum rank i.e. 4. When multiple algorithms give the same value of intracluster distance, the rank given to the algorithms is the average of the ranks obtained by the algorithms in case they result in different values. All the algorithms are ranked accordingly for each dataset (see Table 10). The average rank R_j of the j^{th} algorithm is represented in Eq. (11).

$$R_j = \frac{Sum\ of\ total\ ranks\ obtained\ by\ j^{th}\ algorithm}{Total\ number\ of\ data\ sets} \quad (11)$$

The bracketed values in Table 10 are the ranks for the algorithms. Fig. 2 compares the intracluster distance obtained using KMeans-ALO with that of K-Means, KMeans-PSO, KMeans-FA, DBSCAN and Revised DBSCANfor the 8 data sets mentioned based on the rank obtained.

The Friedman statistics is given by Eq. (12).

$$F_F = \frac{(N-1)X_F^2}{N(k-1)-X_F^2} \quad (12)$$

where, $X_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right]$

here N is the number of datasets; k is the number of algorithms used. The Friedman statistic $F_F$is distributed according to the F-distribution with *(k-1)* and *(k-1) (N-1)* degree of freedom. For 6 algorithms and 8 datasets the degree of freedom is between 5 and 35. Critical value of F (5, 35) for $\alpha = 0.10$ is 2.019 [29]. If the F_F value is less than the critical value then the null hypothesis will be accepted otherwise it will be rejected. For the 6 algorithms and 8 datasets the X_F^2 value is 11.64,278 and the F_F value is 2.8740. As the F_F value is greater than the critical value the null hypothesis is rejected. Thus it can be concluded that there exists some differences among the algorithms considered.

The null hypothesis is rejected; hence Holm's procedure is performed as the post hoc test. The Holm's method determines whether the performance of the control algorithm is statistically better than the other approaches. Here, the null hypothesisH_0 represents the pair of algorithms compared is equivalent. To perform the test, z value is computed using the formula given in Eq. (13). Then the probability p is obtained from the table of normal distribution [30], using the z value. The obtained probability p_i is compared with $\alpha/(k-i)$. In this case KMeans-ALO is the control algorithm. It is clear from Table 11 that, the hypothesis is rejected for all the three cases. Thus it can be determined that KMeans-ALO performs statistically better K Means, KMeans-PSO, KMeans-FA, DBSCAN and Revised DBSCAN algorithms.

$$z = \frac{R_i - R_j}{SE} \quad (13)$$

where, $SE = \sqrt{\frac{k(k+1)}{6N}}$

4. Conclusion

The simplicity and efficiency of K-Means has made it popular for cluster analysis. However the random initialization of centroid positions is a disadvantage of this clustering method. In this work effort has been put forwarded to improve the quality of clustering of KMeans clustering algorithm by integrating Ant Lion Optimization which is a nature inspired optimization technique. The proposed algorithm is implemented on 8 different datasets and the performance of the algorithm is compared based on different performance metrics. Intracluster distance and F-measure are the performance metrics used. To obtain better quality of

clusters the value of intracluster distance should be minimum and the value for F-measure should be maximum. The performance of KMeans-ALO is compared against the performance of K-Means, KMeans-PSO, KMeans-ALO, DBSCAN and Revised DBSCAN. The simulation results validate that KMeans-ALO performs better than the K-Means and the other two hybrid approaches mentioned. The Friedman test shows the existence of significant differences among Kmeans, Kmeans-PSO, Kmeans-FA, Kmeans-ALO, DBSCAN and Revised DBSCAN. Furthermore, Holm test reveals that Kmeans-ALO gives superior performance than K-Means, KMeans-PSO, Kmeans-FA, DBSCAN and Revised DBSCAN. The level of confidence for the statistical analysis is considered as 0.10 which means that accuracy of the results obtained by the proposed Kmeans-ALO is 90%.

Acknowledgements

The authors would like to thank for using the facilities created in Dept. of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, India out of TEQIP-II sponsored seed project entitled “Security Analysis of Cloud Infrastructure” − 2016.

References

[1] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recogn. Lett. 31 (2010) 651−666.
[2] D.P. Kanungo, J. Nayak, B. Naik, H.S. Behera, Hybrid clustering using elitist teaching learning-based optimization: an improved hybrid approach of TLBO, Int. J. Rough Sets Data Anal. (IJRSDA) 3 (2016) 1−19.
[3] J.M. Pena, J.A. Lozano, P. Larranaga, An empirical comparison of four initialization methods for the k-means algorithm, Pattern Recogn. Lett. 20 (1999) 1027−1040.
[4] M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Syst. Appl. 40 (2013) 200−210.
[5] C.D. Wang, J.H. Lai, Energy based competitive learning, Neurocomputing 74 (2011) 2265−2275.
[6] C.D. Wang, J.H. Lai, C.Y. Suen, J.Y. Zhu, Multi-exemplar affinity propagation, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 2223−2237.
[7] C. D. Wang, J. H. Lai, J. Y. Zhu, A conscience on-line learning approach for kernel-based clustering. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 531-540). IEEE..
[8] K. Krishna, M.N. Murty, Genetic K-means algorithm, IEEE Trans. Syst. Man. Cybern. Part B (Cybernetics) 29 (1999) 433−439.
[9] R. F. Abdel-Kader, Genetically improved PSO algorithm for efficient data clustering. In Machine Learning and Computing (ICMLC), 2010 Second International Conference on (pp. 71-75). IEEE..
[10] T. Hassanzadeh, M. R. Meybodi, A new hybrid approach for data clustering using firefly algorithm and K-means. In Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on (pp. 007-011). IEEE..
[11] X. Han, L. Quan, X. Xiong, M. Almeter, J. Xiang, Y. Lan, A novel data clustering algorithm based on modified gravitational search algorithm, Eng. Appl. Artif. Intell. 61 (2017) 1−7.
[12] T. Niknam, B.B. Firouzi, M. Nayeripour, An efficient hybrid evolutionary algorithm for cluster analysis, in: World Applied Sciences Journal, 2008.
[13] Y.T. Kao, E. Zahara, I.W. Kao, A hybridized approach to data clustering, Expert Syst. Appl. 34 (2008) 1754−1762.
[14] C.A. Murthy, N. Chowdhury, In search of optimal clusters using genetic algorithms, Pattern Recogn. Lett. 17 (1996) 825−832.
[15] S. Bandyopadhyay, U. Maulik, An evolutionary technique based on K-means algorithm for optimal clustering in RN, Inf. Sci. 146 (2002) 221−237.
[16] M. Alswaitti, M. Albughdadi, N. Isa, Density-based particle swarm optimization algorithm for data clustering, Expert Syst. Appl. 91 (2018) 170−186.
[17] J. Nayak, D.P. Kanungo, B. Naik, H.S. Behera, Evolutionary improved swarm-based hybrid K-means algorithm for cluster analysis, in: Proceedings of the Second International Conference on Computer and Communication Technologies, Springer, New Delhi, 2016, pp. 343−352.
[18] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67−82.
[19] S. Mirjalili, The ant lion optimizer, Adv. Eng. Software 83 (2015) 80−98.
[20] R. Pradhan, S.K. Majhi, J. Pradhan, B.B. Pati, Performance evaluation of PID controller for an automobile cruise control system using ant lion optimizer, Eng. J. 21 (2015) 347−361.
[21] R. Pradhan, S.K. Majhi, J. Pradhan, B.B. Pati, Antlion optimizer tuned PID controller based on bode ideal transfer function for automobile cruise control system, J. Ind. Inf. Integrat. 9 (2018) 45−52.
[22] M.M. Nischal, S. Mehta, Optimal load dispatch using ant lion optimization, Int. J. Eng. Res. Afr. 5 (2015) 10−19.
[23] R. Pradhan, S.K. Majhi, B.B. Pati, Design of PID controller for automatic voltage regulator system using Ant Lion Optimizer, World J. Eng. 15 (2018) 373−387. https://doi.org/10.1108/WJE-05-2017-0102.
[24] H.M. Zawbaa, E. Emary, C. Grosan, Feature selection via chaotic antlion optimization, PLoS One 11 (3) (2016).
[25] W. Yamany, A.Tharwat, M. F. Hassanin, T. Gaber, A. E. Hassanien, T. H. Kim, A new multi-layer perceptrons trainer based on ant lion optimization algorithm. In Information Science and Industrial Applications (ISI), 2015 Fourth International Conference on (pp. 40-45). IEEE.
[26] K. Bache, M. Lichman, UCI Machine Learning Repository, 2013.
[27] M.K. Pakhira, A linear time-complexity k-means algorithm using cluster shifting, in: Computational Intelligence and Communication Networks (CICN), 2014 International Conference on, IEEE, 2014, pp. 1047−1051.
[28] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1−30.
[29] F Distribution Table, 2018 Mar18. Retrieved from http://www.socr.ucla.edu/applets.dir/f_table.html.
[30] Normal Distribution Table. Retrieved from http://math.arizona.edu/~rsims/ma464/standardnormaltable.pdf.

[31] M. Ester, H.P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, vol. 96, 1996, pp. 226–231.

[32] T.N. Tran, K. Drab, M. Daszykowski, Revised DBSCAN algorithm to cluster data with dense adjacent clusters, Chemometr. Intell. Lab. Syst. 120 (2013) 92–96.

[33] S. Das, T. Deb, N. Dey, A.S. Ashour, D.K. Bhattacharya, D.N. Tibarewala, Optimal choice of k-mer in composition vector method for genome sequence comparison, Genomics 110 (2018) 263–273.