Journal of King Saud University – Computer and Information Sciences 31 (2019) 52–61

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

FSDE-Forced Strategy Differential Evolution used for data clustering

Meera Ramadas [a,*], Ajith Abraham [b], Sushil Kumar [a]

[a] Amity University, Noida, Uttar Pradesh, India
[b] MIR Labs, Washington, USA

ARTICLE INFO

Article history:
Received 6 June 2016
Revised 27 November 2016
Accepted 15 December 2016
Available online 25 December 2016

Keywords:
Mutation
Crossover
Centroid
Cluster quality
Quantization error
Index

ABSTRACT

Differential evolution algorithm has seen various changes through numerous researches. Performance of the various algorithms depends on the changes in mutation and crossover strategies. Here in this paper, we are proposing a new variant of differential evolution named Forced Strategy Differential Evolution (FSDE), by creating a new mutation strategy. This strategy uses two parameters for mutation: a constant parameter and a variable parameter. FSDE will be applied on clustering using the k means technique. Experiments were conducted for various standard benchmark functions. FSDE was compared with the classical DE, GA and PSO in the field of clustering and the cluster quality results are tabulated. The results obtained show that the strategy implemented is more efficient than the other mutation strategies.

1. Introduction

With the ever evolving and rapid progress in the area of research, the amount of data that needs to be processed or produced has increased multi fold. With this increase in the data collection arise major challenges in the field of data mining. Data mining or knowledge discovery is the technique of identifying the data from various aspects and then properly categorizing these data information. The data mining software analyses the relationships and patterns that exist within the data. Traditionally data mining technique has been classified under two types namely supervised learning and unsupervised learning. Supervised learning technique is used for classifications and predictions whereas unsupervised learning techniques are those in which no predictions or classifications are possible. Clustering of data is an unsupervised learning technique. Cluster analysis is an emerging area of interest to many researchers in the field of data mining. Clustering divides data into useful and meaningful groups called clusters. The main aim of clustering is that the objects or data in a group should be similar to one another and dissimilar to the objects in the other group. Clustering determines the intrinsic grouping within a collection of unlabelled data. It is a major method for statistical data analysis to be used in the field of machine learning, pattern recognition, image processing, data compression and information retrieval. Jain (2010) has summarized clustering technique and the various methods used. Clustering of data plays a vital role in efficient data mining, voice recognition, web mining, market analysis etc. Fast and accurate clustering of data plays an important role in the field of automatic information retrieval system. It is considered as a multi objective optimization problem. It involves an iterative task of trial and error. Clustering can be classified as hard clustering and soft clustering. In hard clustering, each object belongs to one cluster or does not belong to any cluster. In soft clustering or fuzzy clustering, each object may belong to more than one cluster (Bezdek et al. (1984)). Clustering algorithm can be categorized into hierarchical and partitional algorithms. In hierarchical clustering, a hierarchy of partitions is constructed and a dendogram representation is created. In this technique, each partition is grouped within the partition of next level in the hierarchy. In partitional clustering, a single partition is constructed with a given number of non-overlapping clusters. The main disadvantage of partitional clustering is to find partition of data with a specified number of clusters which minimizes within cluster differences. Partitional algorithms are iterative and usually converge to local minima.

The simplest and most popular partitional clustering algorithm is the k-means technique which was coined by MacQueen (1967).

* Corresponding author.
E-mail address: meera_mgr@rediffmail.com (M. Ramadas).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2016.12.005
1319-1578/

Here a given set of *N* data are partitioned into k different clusters. Grouping of data is done by minimizing the Euclidian distance between the data and centroid. It is one of the simplest unsupervised learning algorithms used for clustering. This algorithm is significantly sensitive to initial randomly selected centroid. Result of k means algorithm depends on the initial mean values and so frequently suboptimal partitions are found. As k means algorithm may converge to suboptimal partitions, some stochastic optimization approach is used to avoid this situation and thereby to find a globally optimum solution. Such problems can be solved using the evolutionary algorithm. Coello et al. (2002) stated that evolutionary algorithm works in a robust and efficient manner for clustering. Evolutionary algorithm, which is a part of evolutionary computing, uses biological methods of reproduction, recombination, mutation and selection. Storn and Price (1997) introduced differential evolution (DE) algorithm which follows the concepts of the evolutionary algorithm. DE is simple, stochastic algorithm based on the population that help to solve optimization problems. The effectiveness and performance of DE is determined by the control parameters and test vector generation strategy. Many variants are designed by changing these strategies and control of test vector parameters.

The aim of this paper is to present a variant of differential evolution approach .This approach is then combined with k-means to be applied on to a clustering problem. The first section of the paper discusses the basic clustering approach. The second section deals with the explanation on differential evolution technique. The third section introduces the readers to the variant of DE algorithm named as Forced Strategy Differential Evolution (FSDE). The experimental results obtained from FSDE are explained in the successive sections. Then a detailing is done on the implementation of FSDE and other evolutionary algorithms on k means approach for applying on data clustering.

2. Related works on evolutionary algorithms in clustering

Applying the concept of evolutionary computing to clustering problem has been the topic of research for a long time. Numerous variants of evolutionary algorithms like DE, GA, PSO were created and these variants were applied to the clustering problems. Paterlini and Thiemo (2004) gave a performance comparison on genetic algorithm (GA), PSO and differential evolution (DE) for a medoid evolutionary clustering approach. The results showed that DE approach was far superior compared to GA and PSO and that DE should be considered over the other algorithms for clustering. Zaharie (2005) studied the applicability of crowding differential evolution to unsupervised clustering. This approach allows the identification of clusters of arbitrary shapes using multi centre descriptions for them. Abraham et al. (2006) describes a novel approach for partitioning text document into clusters using an improved version of classical differential evolution. A modified mutation scheme was introduced to improve convergence properties. This modified DE was then used for clustering text document for retrieving important information. A new validation index was also proposed for high dimensional document clustering problems by modifying the CS measure. This technique was shown to be superior in speed and quality of clustering. Zhang and Sanderson (2007) showed that implementation of DE is mostly based on crossover probability and mutation factor. Changes done to these parameters will affect the performance of DE.

Zhang et al. (2008) also proposed an advanced PSO and differential evolution method for spatial clustering with obstacle constraints (SCOC). The proposed method showed better quantization error and constringency speed. Indrajit et al. (2009) proposed an application of differential evolution to fuzzy clustering for categorical data sets. The proposed algorithm effectively optimizes the fuzzy c-medoids error function globally. Maulik and Indrajit (2010) devised a modified differential evolution (DE) based-fuzzy c-mediod (FCMdd) clustering of categorical data set. This technique shows the superiority of integrated clustering and supervised learning approach. Maulik et al. (2010) also proposed a new real – coded modified differential evolution based automatic fuzzy clustering algorithm that automatically calculates the number of clusters and the proper partition from a data set. In this paper, the assignment of points to different clusters is based on a Xie-Beni index which considers the Euclidian distance.

Alguliev et al. (2011) proposed a document summarization model which separates the key sentences from the given document while removing the redundant information in the summary. The results show that the proposed method was superior to the earlier summarization models. Pham et al. (2011) introduced a new approach to cluster datasets of mixed data type. RANKPRO (random search with k prototype) combined the bees algorithm with the k prototype. RANKPRO algorithm proved to be more efficient than the k prototype approach. Suarez-Alvarez et al. (2012) introduced a unified statistical approach to normalize all attributes of mixed datasets. Clustering of several standard datasets is also performed in this paper. Qu et al. (2012) gave a neighbourhood mutation strategy and combined it with various niching differential evolution (DE) algorithms to solve multimodal optimization problems. This technique has faster convergence with higher accuracy. The mutation strategy in this technique was able to generate a stable niching behaviour and was able to locate and maintain multiple global optima.

Hatamlou (2013) devised a new heuristic method inspired from the black hole phenomena. The experimental results showed that the technique outperformed the existing classical methods. This method was applied to the field of clustering. Saha and Bandyopadhyay (2013) devised a new multiobjective (MO) clustering technique (GenClustMOO) which can automatically partition data into appropriate clusters. The effectiveness of the method was compared against k means and single linkage method. Singh and Saha (2014) gave a solution to clustering after analysing and removing the drawbacks of Euclidean distance and point symmetry based distance measures and merging the improved versions into one method to get best of both methods. This method speeds up the computation time. Thein et al. (2015) proposed differential evolution for clustering and compares its purity with k-means algorithm. The results were tested on medical datasets of Pima, Liver and Heart from UCI data repository. According to the results obtained, DE outperformed the k means algorithm for medical datasets. This work shows that DE performs better when robust clustering is needed. This work also eliminates the disadvantages of mean technique. Mukherjee et al. (2016) gave a modified version of differential evolution for solving dynamic optimization problems (DOP) efficiently. The algorithm was named as Modified DE with Locality induced Genetic Operators (MDE-LiGO) and it integrates changes in three stages of classical DE framework. Wu et al. (2016) devised a multi-population based approach to achieve a unit of multiple strategies. The resulting new variant named multi-population ensemble DE (MPEDE) consists of three mutation strategies. Ramadas et al. (2016) introduced a strategy a hybrid technique of differential evolution and Flower Pollination Algorithm (ssFPA/DE). The results were tabulated and efficiency of the new approach was justified.

3. Data clustering approach

Clustering is of two types: hard clustering and soft clustering. In hard clustering, each data is a member of one cluster. In soft clus-

tering or fuzzy clustering, each data may belong to multiple clusters.

One of the most popularly used hard clustering approaches is the k means algorithm. The standard algorithm for k- means was proposed by Stuard Lloyd in 1982. The aim of this algorithm is to find the best partition of n entities into k groups or clusters in such a way that the total distance between the group members and its centroid are minimized. It iterates between updating the assignment of data to clusters and updating cluster's summarization or centres. In this technique, k centroids are initialized. These centers are chosen at random initially. Then, each point is assigned to the nearest centroid by calculating the Euclidean distance between the centroid and the data point. Euclidean distance is the geometric distance in the multidimensional space. K means algorithm always use Euclidean distance to calculate the closeness between the centroid and data. It is computed as:

$$\text{distance}(x,y) = \left\{ \sum_i (x_i - y_i)^2 \right\}^{\frac{1}{2}} \quad (1)$$

where x_i, y_i are the two points between which the distance has to be calculated. Now new centers are calculated for each group. If there is a change in the centroid assignment, again the distance from each data to the new centroid is calculated. These are iteratively performed until there is no change in the centroid. The flowchart for data clustering using k means algorithm is depicted in Fig. 1. Suppose we have k clusters where c = 1, 2, … k. Let the data set be X where $X = x_1, x_2, \ldots x_n$ and set of centers be denoted as V where $V = V_1, V_2, \ldots V_c$. The centroid of the cluster is calculated as:

$$V_i = 1/c_i \sum_{j=1}^{c_i} x_j \quad (2)$$

where c_i is the number of data points in the ith cluster.
The basic algorithm for k means is given as below:

Select k points as the initial centres.
Repeat until centres do not change:
Form k clusters by grouping data near the centroid to its corresponding cluster.
Recalculate the centroid for each cluster.
End loop.

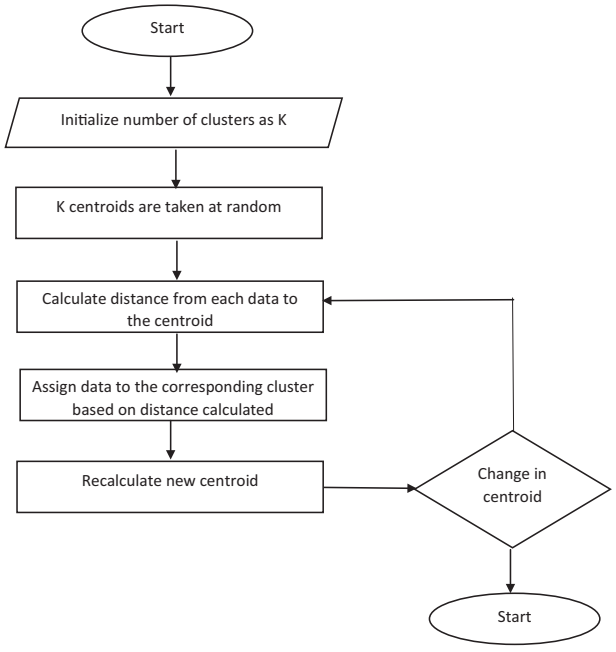

Figure 1. Flowchart for k means algorithm.

K means algorithm is easy to implement and computationally fast. Fong et al. (2014) stated the main drawback of this technique is that there is no surety of finding global optima. Classical k means algorithm randomly assigns initial point and usually finds a local optimum clustering result. So, it needs a global optimized algorithm to remove the defects of k-means technique. Differential evolution technique is a simple heuristic technique for global optimization proposed by Storn and Price (1997). DE algorithm has strong global search ability, higher accuracy and slower convergence speed while k means clustering algorithm has faster convergence speed. Combining k means algorithm with differential evolution stabilizes the local and global search. K means algorithm finds its application in various fields of feature learning, cluster analysis and vector quantization.

4. Differential evolution algorithm

DE uses a population of NP candidate solutions denoted as $X_{i,G}$ where $i = 1, 2 \ldots NP$ where index i denote population and G represents generation of the population. Differential evolution algorithm depends on three operations mainly mutation, selection and reproduction.

Mutation: This operator makes DE different from other Evolutionary algorithms. It computes the weighted difference between the vectors in population. Mutation process starts by selecting three individuals at random from the population. This operation extends the workspace. For a given parameter $X_{i,G}$ we are randomly selecting 3 vectors $X_{r1,G}, X_{r2,G}$ and $X_{r3,G}$ such that r_1, r_2, r_3 are distinct. Then the donor vector $V_{i,G}$ is computed as:

$$V_{i,G} = X_{r1,G} + F \times (X_{r2,G} - X_{r3,G}) \quad (3)$$

where, $i = 1 \ldots NP$, $r_1, r_2, r_3 \in \{1, \ldots, NP\}$ are randomly selected and satisfy: $r_1 \neq r_2 \neq r_3 \neq i$, $F \in [0, 1]$, F is the control parameter for mutation proposed by Storn and Price (1997). Here F is a constant from [0, 1]. Mutation function demarcates one DE scheme from another.

Crossover: This process also called as recombination incorporates successful solutions into the population. The trial vector $U_{i,G}$ is created for the target vector $X_{i,G}$ through binomial crossover. Elements of donor vector enter trial vector with probability $C_r \in [0, 1]$. C_r is the crossover probability which is selected along with population size $NP \geqslant 4$.

$$U_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & \text{if } rand_{i,j}[0,1] \leqslant C_r \text{ or if } j = I_{rand} \\ X_{j,i,G+1} & \text{if } rand_{i,j}[0,1] > C_r \text{ or if } j \neq I_{rand} \end{cases} \quad (4)$$

Here rand $_{i,j} \approx \cup[0, 1]$ and I_{rand} is random integer from 1,2 … N.

Selection: This operation differs from the selection operation of other evolutionary algorithms. Here the population for next generation is selected from vectors in current population and its corresponding trial vectors .The target vector $X_{i,G}$ is compared with the trial vector $V_{i,G}$ and the lowest function value is taken into next generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if } f(U_{i,G+1}) \leqslant f(X_{i,G}) \quad \text{where } i = 1, 2, \ldots N \\ X_{i,G} & \text{otherwise} \end{cases} \quad (5)$$

Mutation, crossover and selection operations are continued until some stopping criteria are reached.

5. Proposed variant of differential evolution

A new strategy has been proposed for mutation called FSDE. As it involves the best solution vector $X_{G,best}$, it coincides faster as compared to the traditional strategies which has only random vec-

tors. Here, two control parameters are being used. The parameter F known as mutation factor takes a constant value while the new parameter N takes a varying value which lies between (0,1).The parameter F uses the constant value of 0.6 in our proposed technique so that the value of donor vector lies between the allowable range. As we are taking two different control parameters, the value of donor vector is improved greatly and hence the efficiency of DE algorithm is enhanced profoundly. The proposed strategy is given as:

$$V_{i,G} = X_{r1,G} + N.((X_{G,best} - X_{r2,G}) - F.(X_{G,best} - X_{r3,G})) \quad (6)$$

Here the random selection of base vector prevents the strategy from becoming greedy in nature.

6. Test problem

The above stated hybrid algorithm FSDE was implemented on i7 core processor, 64 bit operating system with 12 GB RAM using MATLABr2008b and a comparative result was obtained with the original DE algorithm. The traditional mutation strategies were replaced with the proposed mutation strategy and FSDE was composed. In the experiment conducted, mutation constant F is given the value 0.6 and the crossover probability C_r is given the value 0.8. We have taken fifteen different functions and calculated the results by fixing the value to reach and number of iterations. The maximum number of iterations is fixed as 5000 and the maximum number of evaluations as 5,000,000. The value to reach (VTR) is the global minimum or maximum of the function or it is a value to stop the optimization if it is reached. We have tabulated various results by fixing the dimension as 25 and 50. The results are tabulated for comparison with the existing algorithms. The column named significance shows if the result obtained for FSDE is better than the other strategies. Few of the results obtained are given in Table 1.

Based on Best Value (vtr = 1.e−015):

A comparative analysis was performed and study was done on each technique. By setting the value-to-reach (VTR) as e−015 and dimension as 25 and 50, the best value, number of function evaluation (NFE) and the CPU time of different function strategies are calculated. In some functions, the results are good for both classical DE and proposed algorithm. The overall result obtained shows that the FSDE approach is performing better than the classical DE approach. The Friedman statistical test runs are conducted on FSDE algorithms to validate the results. Based on the values from Table 1, the Friedman test was applied and the results are tabulated in Table 2. The ranks obtained after the Friedman test is tabulated in Table 3.

Table 2
Test statistics using Friedman's test.

N	25
Chi sq	23.6
Df	5
Asymptotic Significance	0.0003

Table 3
Ranks of the diverse strategies.

Strategies	Mean rank on best value
DE/best/1	2.6
De/rand/1	3.2
DE/best-to-rand/1	2.8
De/best/2	4.3
DE/rand/2	5.1
FSDE	2.9

Table 1
Best value for different functions.

Function	D	VTR	DE						Significance
			DE/best/1	DE/rand/1	DE/best-to-rand/1	De/best/2	DE/rand/2	FSDE	
Sphere	50	1.e−015	9.73e−016	6.9e−016	7.532e−016	9.655e−016	7.17e+0	**6.04e−016**	+
	25	1.e−015	9.34e−015	9.35e−015	9.539e−015	9.42e−015	6.92e+000	**8.9e−016**	+
Beale	50	1.e−015	3.265e−016	**2.318e−016**	3.713e−016	7.587e−016	7.725e−016	5.95e−016	–
	25	1.e−015	4.26e−015	7.72e−015	1.125e−015	1.36e−017	7.5e−015	3.6e−016	–
Booth	50	1.e−015	3.497e−016	**2.0514e−016**	6.0738e−016	7.0792e−016	8.35e−016	3.28e−016	–
	25	1.e−015	1.807e−015	7.55e−016	1.95e−015	2.75e−015	6.47e−015	9.4e−016	–
Schwefel	50	**1.e−015**	**−1.8e+003**	**−2.253e+003**	**−7.8403e+001**	**−1.38e+003**	**−1.66e+003**	**−4.56e+003**	NA
	25	**1.e−015**	**−4.22+002**	**−4.8e+002**	**−1.67e+003**	**−4.47e+003**	**−1.5e+003**	**−2.5e+003**	NA
Michlewicz	50	**1.e−015**	**−7.6399e+00**	**−7.214e+00**	**−7.39e+00**	**−6.959 e+00**	**−6.847 e+00**	**−7.34e+00**	NA
	25	1.e−015	−7.69e+00	−7.64e+00	−6.87e+00	−7.35e+00	−6.98 e+00	**−7.1e+00**	NA
Schaffer N.2	50	1.e−015	6.6e−016	8.88e−016	4.43e−016	6.55e−016	8.87e−016	**2.22e−016**	+
	25	1.e−015	1.33e−015	1.33e−015	6.66e−016	5.3e−015	1.33e−015	0	–
Schaffer N.4	50	1.e−015	3.05e−015	2.9e−001	2.92−001	2.93e−001	2.89e−001	**2.82e−001**	+
	25	1.e−015	2.92e−001	2.92e−001	2.92e−001	2.92e−001	2.92e−001	**2.92−001**	NA
HimmelBlau	50	1.e−015	1.6e−016	8.05e−016	3.83e−016	9.12e−016	**1.46e−016**	3.35e−016	–
	25	1.e−015	4.83e−015	4.42e−015	1.902e−015	3.95e−015	5.14e−015	1.59e−015	+
Bird	50	1.e−015	**−1.035e+002**	**−1.067e+002**	**−1.05e+002**	**−1.065e+002**	**−1.03e+002**	**−1.029e+002**	NA
	25	1.e−015	**−9.303e+001**	**−1.04e+002**	**−1.066e+002**	**−1.034e+002**	**−1.04e+002**	**−1.06e+002**	NA
Extended cube	50	1.e−015	3.31e−015	4.98e−005	6.1e−008	1.93e−005	2.68e+00	5.46e−008	+
	25	1.e−015	5.701e−008	5.212e−005	7.1003e−008	1.73e−005	2.92e+009	5.86e−007	–
Ackeley	50	1.e−015	7.19e−015	6.46e−012	7.99e−015	3.63e−013	3.09e+00	7.99e−015	–
	25	1.e−015	7.99e−015	5.02e−015	7.99e−015	3.59e−013	3.213e+00	**4.4e−015**	+
Gold	50	1.e−015	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**	NA
	25	1.e−015	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	3.00e+00	NA
Griewank	50	1.e−015	9.99e−016	9.99e−016	1.6e−013	6.56e−013	1.07e+00	7.77e−015	–
	25	1.e−015	**1.477e−002**	9.214e−015	7.88e−015	5.07e−009	1.06e+00	9.9e−016	–
Rastrigin	50	1.e−015	**1.79e+001**	**1.23e+002**	**7.47e+001**	**1.28e+002**	**1.52e+002**	**2.98e+001**	NA
	25	1.e−015	**3.61e+001**	**1.181e+002**	**8.17e+001**	**1.727e+002**	**1.674e+002**	**0**	NA
Rosenbrock	50	1.e−015	9.6e−016	**1.07e−008**	7.88e−016	3.9e−009	1.07e+005	1.107e+001	–
	25	1.e−015	3.98e+00	1.403e−008	6.9e−015	**1.56e−011**	7.15e+004	8.5e−016	–

The values highlighted show the overall best values of each function.

7. Proposed variant in clustering

This section discusses the implementation of the mutation variant of DE in clustering using the k means algorithm. Each record in a dataset is handled as random sample of population under consideration. Now consider that these datasets are clustered to k random groups. Partitions of the data set are carried out on the basis of certain objective functions. This is a feature that opts to an optimization problem to minimize or maximize the function from a set of given available alternatives. This function determines how well the chosen solution performs. The fitness of each solution is performed by calculating the squared error function between centroid and entity point which is defined as:

$$\text{Fitness}(C) = \sum_{j=1}^{k}\sum_{i=1}^{n} ||x_i^j - c^j||^2 \quad (7)$$

where x_i^j is the entity point, c^j is the centroid and $||x_i^j - c^j||$ gives the distance between the centroid and the entity point. Then for each k groups, centroid is selected arbitrarily. Using Euclidean distance method, distance is calculated for each data from its corresponding centroid in the group. K-means algorithm terminates if there is no change in centroid allocated. Result of the k-means algorithm is used as one of the elements of DE algorithm while the rest of the elements are initialized randomly. FSDE algorithm performs the proposed mutation and crossover function. If the resultant value obtained has better cost function, then the resultant value replaces the least fit value in the population. The FSDE algorithm terminates if the maximum iteration is exceeded. Fig. 2 shows the flow chart for the clustering technique using the variant of DE.

8. Experimental results on clustering

Here, the experiment was conducted on five standard datasets with numeric data to compare the performance of the k means algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and classical DE with FSDE in clustering. The k means algorithm of clustering was incorporated with GA, PSO, classical DE and DE with FSDE for performing the data clustering. The clustering algorithms were implemented on i7 core processor, 64 bit operating system with 12 GB RAM using MATLABr2008b and a comparative result was obtained for the various algorithms. The corresponding cluster graph and curve graph for each dataset were obtained. The cluster quality of the clusters obtained was compared. Five real time datasets from UCI repository of machine learning database Blake et al. (1998) are used. The datasets used are described below:

(a) Fisher Iris dataset ($n = 150$, $d = 4$, $k = 3$): This is a standard dataset with 150 inputs for 3 different flower types: sentosa, virginica and versicolour. Here 4 different features of flower are measured: type, petal width, sepal width and sepal length.
(b) Morse dataset ($n = 1296$, $d = 5$, $k = 4$): This dataset consists of 36 rows and 36 columns representing the morse code for letters from A–Z and numbers from 0 to 9. Each letter or num-

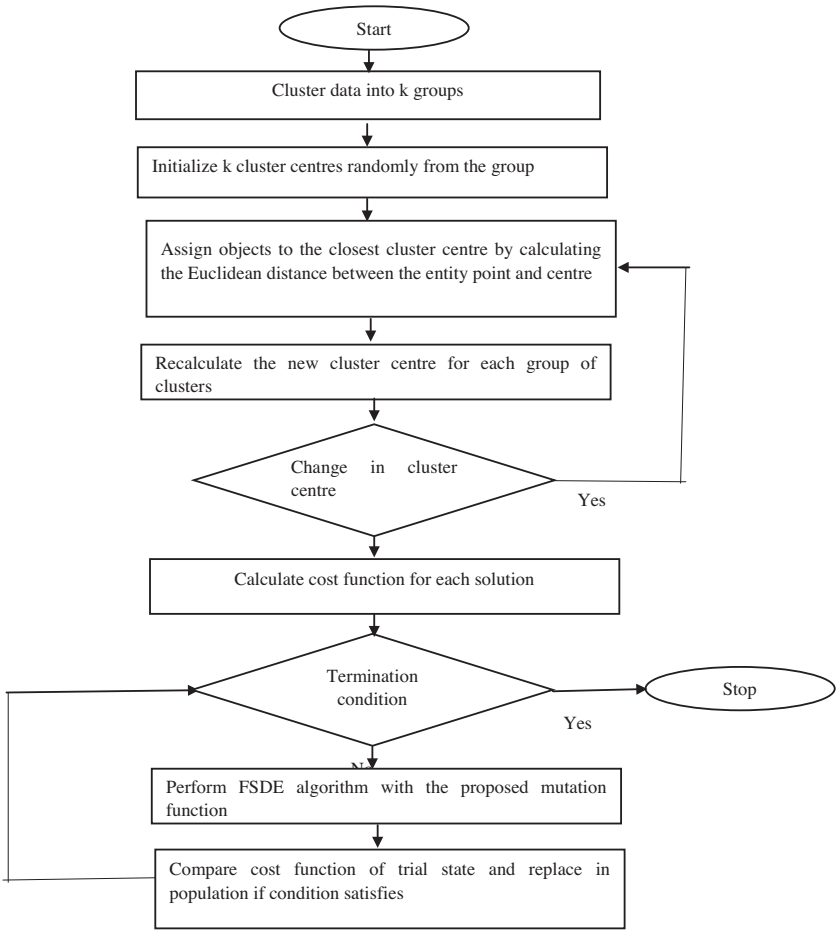

Figure 2. Flow chart for variant of DE in clustering.

ber is represented with a dash or dot. Morse code consists of 5 elements: short mark, longer mark, intra character gap, short gap and medium gap.

(c) Hogg dataset ($n = 30$, $d = 6$, $k = 4$): This dataset is a lab based report of bacteria counts in different shipment of milk. Here, 6 sets of bacteria count are taken from 5 different shipments of milk.

(d) Weather dataset ($n = 60$, $d = 5$, $k = 4$): This is an unsupervised dataset obtained from lab test. It describes main characteristics of weather database for 5 different attributes: outlook, temperature, humidity, windy and play.

(e) Stock dataset ($n = 950$, $d = 10$, $k = 4$): This is a real dataset obtained from simulated stock returns. Here the stock returns for 10 different companies are being considered in this dataset.

These datasets were used as input for clustering and results were obtained for various algorithms under consideration. The cluster graph and curve graph for the iris data set have been given below. Figs. 3–6 shows the cluster obtained after performing genetic algorithm, PSO, DE and FSDE respectively for the iris dataset. The x-axis shows the petal length and y-axis shows the petal width of the iris data set. Figs. 7–10 shows the curve graphs obtained during clustering using genetic algorithm, PSO, DE and FSDE for the iris dataset. In these graphs, the x-axis shows the number of iterations and y-axis shows the best cost obtained at each iteration.

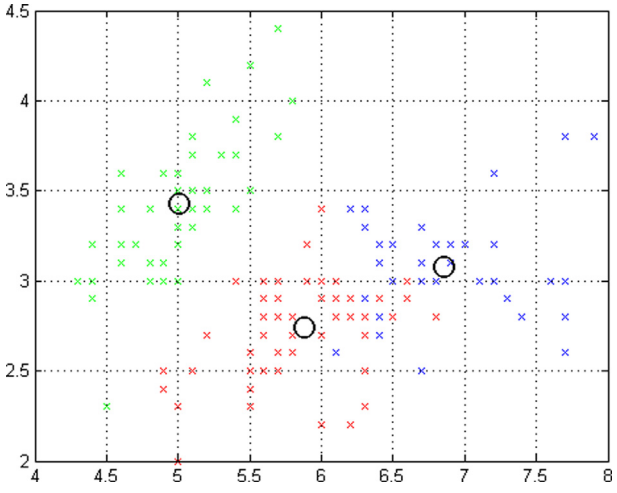

Figure 3. Clusters after applying genetic algorithm.

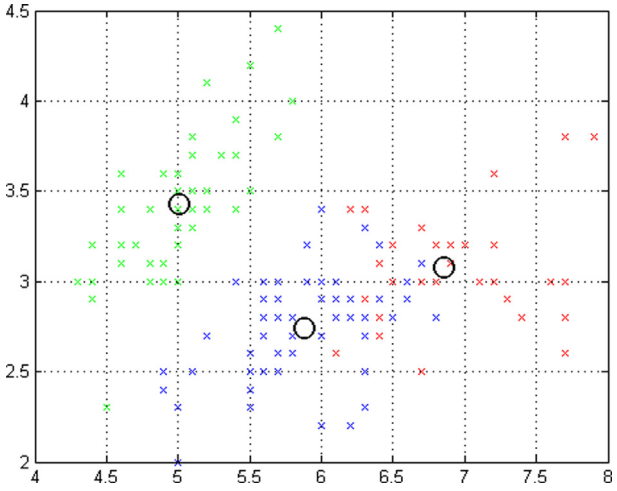

Figure 4. Clusters after applying PSO.

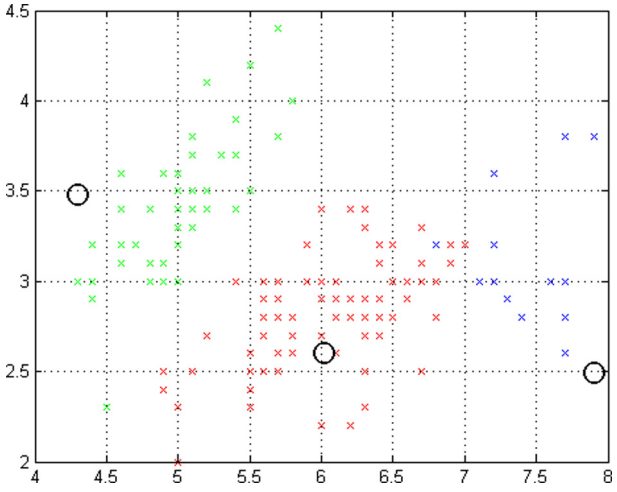

Figure 5. Clusters after applying classical DE.

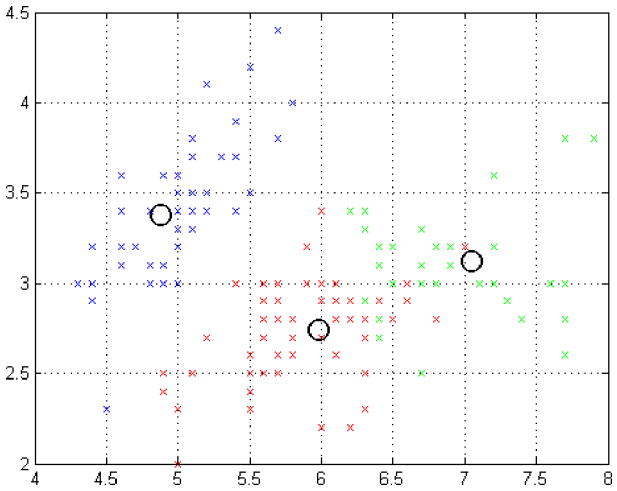

Figure 6. Clusters after applying FSDE algorithm.

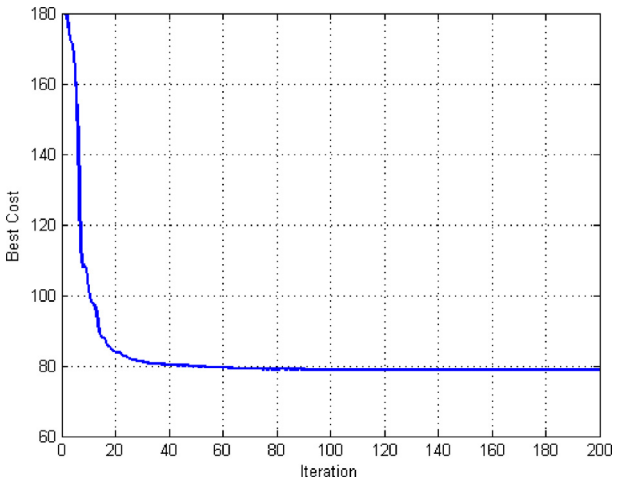

Figure 7. Curve graph for genetic algorithm.

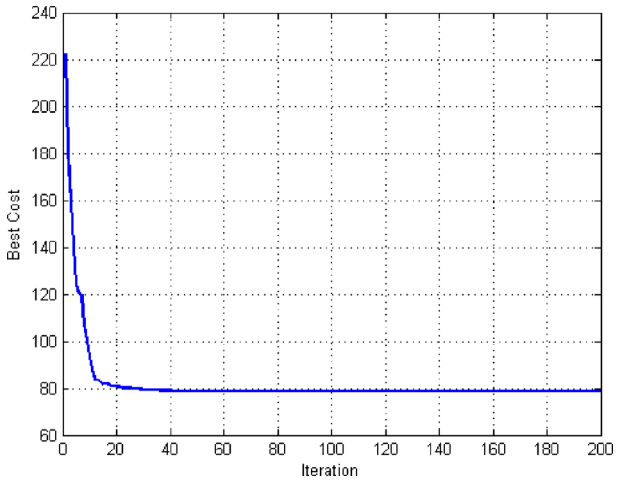

Figure 8. Curve graph for PSO.

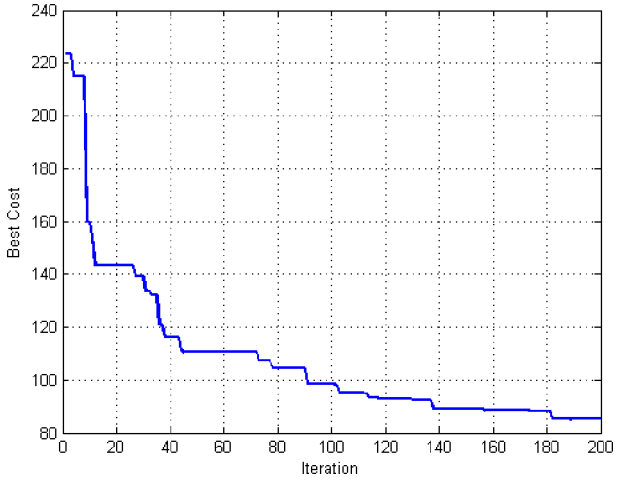

Figure 9. Curve graph for FSDE.

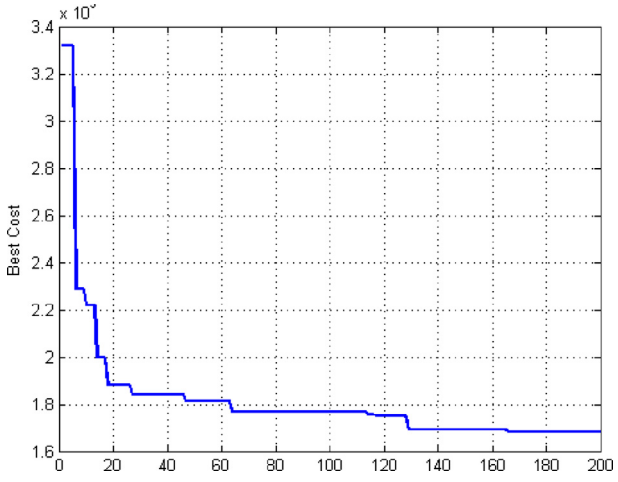

Figure 10. Curve graph for classical DE.

8.1. Cluster quality

Quality of the clusters obtained can be compared based on the following parameters:

8.1.1. Intra cluster distance

This calculates the distance of the members in the cluster. It is calculated as the sum of squares of distance from the members of each cluster to its centroid. Minimum intra cluster distance gives good clusters. The formula for intra cluster distance is given as:

$$intra = \sum_{j=1}^{k} x_i^j - c_j^2 \tag{8}$$

where $x_i^j - c^j$ is the distance between particles and centroid .Lower the intra cluster distance value, better the cluster formed. The comparative results obtained for mean of intra cluster distance is given in Table 4.

8.1.2. Inter cluster distance

It calculates the distance of all the pairs of centroid of different clusters. It is the sum of squares of distance between each cluster centroid. The formula is given as:

$$inter = \min(c_i - c_j)^2 \tag{9}$$

Here c_i and c_j are the centroids of cluster i and j. Maximum inter cluster distance, better the cluster formed. The comparative results obtained for mean of inter cluster distance of the various algorithms is given in Table 5.

8.1.3. Quantization error

Vector quantization divides large set of data into clusters having almost same number of points closest to them. Goal of vector quantization is to reduce the average quantization error. The formula for quantization error Q_e is given as:

$$Q_e = \sum_{j=1}^{k} \left[\sum_{i=1}^{n} ||x_i^j - c^j||^2 / N_j \right] \Big/ k \tag{10}$$

where, c^j is centroid of cluster j, N_j is number of particles of cluster j, $||x_i^j - c^j||$ is the distance between particles and centroid.

Lower the quantization error, better is the cluster formed. The result obtained for quantization error is tabulated in Table 6.

8.1.4. Execution time

It is the total time taken for the execution of task. Lower the execution time, better the cluster. Execution time for the various algorithms is shown in Table 7.

8.2. Validation index

There are various quantitative evaluation techniques available to test the cluster quality and these are known as validation index. It is used as a tool by researchers to test the cluster result. Internal quality compares different set of clusters without reference to external knowledge. A good clustering technique has high within cluster similarity and low inter cluster similarity. Here, we will be calculating two validation indexes: Davies Bouldin (DB) index and Calinski Harabasz (CH) index.

8.2.1. Davies Bouldin (DB) index

It is a matrix for evaluating the cluster algorithm. It is a function of ratio of sum of intra- distances to inter distances (Davies and Bouldin, 1979). If $R_{i,j}$ be the measure of clustering scheme, $M_{i,j}$ is the separation between i and j cluster and S_i is the within cluster scatter for cluster I, then the DB index is defines as the ratio of S_i and $M_{i,j}$ which follows the following properties:

1. $R_{i,j} \geqslant 0$.
2. $R_{i,j} = R_{j,i}$

Table 4
Comparative table for mean intra cluster distance.

Data sets	Mean intra cluster distance				
	K means	GA	PSO	Classical DE	FSDE
Morse ($k = 4$)	16.01	16.17	19.467	18.803	16.07
Iris ($k = 3$)	43.7	48.24	45.78	22.24	20.34
Hogg ($k = 4$)	120.1	122.3	146.66	107.08	99.04
Weather ($k = 4$)	169983.2	170500.6	181200.7	170400.84	169772.8
Stockreturns ($k = 4$)	2011	2134	2345.7	2256.8	1693.03

Table 5
Comparative table for mean inter cluster distance.

Data sets	Mean inter cluster distance				
	K means	GA	PSO	Classical DE	FSDE
Morse ($k = 4$)	230.12	224.14	159.12	241.9	277.55
Iris ($k = 3$)	3851.3	4015.72	4037.6	4144.86	5781.3
Hogg ($k = 4$)	4245.2	4542.57	3158.7	4527.9	4549.71
Weather ($k = 4$)	34567834.2	36725821.4	30251191.2	36689042.51	36798677.6
Stockreturns ($k = 4$)	12876.22	11660.32	18330.7	13,206	26,889

Table 6
Comparative table for quantization error.

Data sets	Quantization error				
	K means	GA	PSO	Classical DE	FSDE
Morse ($k = 4$)	4.03	4.04	4.86	4.7	4.01
Iris ($k = 3$)	14.03	16.08	15.25	7.41	6.78
Hogg ($k = 4$)	28.7	30.57	36.65	26.7	24.76
Weather ($k = 4$)	42567.2	42625.15	45300.17	42600.2	42443.2
Stockreturns ($k = 4$)	530.3	533.5	586.4	564.2	423.2

Table 7
Comparative table for execution time.

Data sets	Execution time				
	K means	GA	PSO	Classical DE	FSDE
Morse ($k = 4$)	14.8	25.1	24.3	17.3	15.46
Iris ($k = 3$)	12.4	25.018	15.34	25.03	14.33
Hogg ($k = 4$)	24.1	25.54	26.34	26.77	24.7
Weather ($k = 4$)	46.3	55.34	56.67	60.65	48.7
Stockreturns ($k = 4$)	14.32	16.45	15.56	36.53	15.32

3. When $S_i \geqslant S_j$ and $M_{i,j} = M_{i,k}$ then $R_{i,j} > R_{i,k}$.
4. When $S_i = S_j$ and $M_{i,j} \leqslant M_{i,k}$ then $R_{i,j} > R_{i,k}$.

Here, lower the value of DB index, better the separation and closeness of the data inside the cluster. The formula for DB index is given as:

$$DB = \frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}\left\{\frac{d(x_i) - d(x_j)}{d(c_i, c_j)}\right\} \tag{11}$$

where, k is number of clusters, i, j are cluster labels, $d(x_i)$ and $d(x_j)$ are all samples in cluster I and j to respective cluster centroids $d((c_i, c_j))$ is the distance between these centroids.

The comparative results obtained for the DB index for the various algorithms are tabulated in Table 8.

8.2.2. Calinski Harabasz (CH) index

It is another method for calculating cluster quality. It is used to evaluate the optimal number of clusters (Caliński and Harabasz, 1974). Higher the value of CH index, the better the cluster that is formed. The formula for CH index is given as:

$$\mathrm{CH} = \frac{trace(S_B)}{trace(S_W)} \cdot \frac{n_p - 1}{n_p - k} \tag{12}$$

where S_B is between cluster scatter matrix, S_W is the internal scatter matrix, n_p is the number of clustered samples and k is the number of clusters. The comparative results obtained for the CH index is given in Table 9.

8.3. Graphical representation

The above tabulated values of Cluster Quality and Validation index have been depicted graphically. Figs. 11 and 12 depict the curve graphs that show the performance curve for execution time in cluster quality and CH index in validation index. The x-axis represents the different datasets used and y- axis represents the value obtained. The line graph shows that the values obtained for the FSDE is significantly better than the values obtained from classical

Table 8
Comparative table for DB index.

Data sets	DB index				
	K means	GA	PSO	Classical DE	FSDE
Morse ($k = 4$)	1.20	1.22	1.19	1.23	1.12
Iris ($k = 3$)	0.63	0.78	0.68	0.64	0.61
Hogg ($k = 4$)	0.432	0.43	0.45	0.41	0.40
Weather ($k = 4$)	0.41	0.45	0.38	0.35	0.34
Stockreturns ($k = 4$)	3.24	4.34	4.23	4.54	2.16

Table 9
Comparative table for CH index.

Data sets	CH index				
	K means	GA	PSO	Classical DE	FSDE
Morse ($k = 4$)	0.78	0.67	0.67	0.64	1.065
Iris ($k = 3$)	0.32	0.22	0.301	0.21	0.34
Hogg ($k = 4$)	0.17	0.168	0.23	0.20	0.26
Weather ($k = 4$)	0.144	0.146	0.133	0.143	0.148
Stockreturns ($k = 4$)	2.12	2.0484	2.01	1.075	2.48

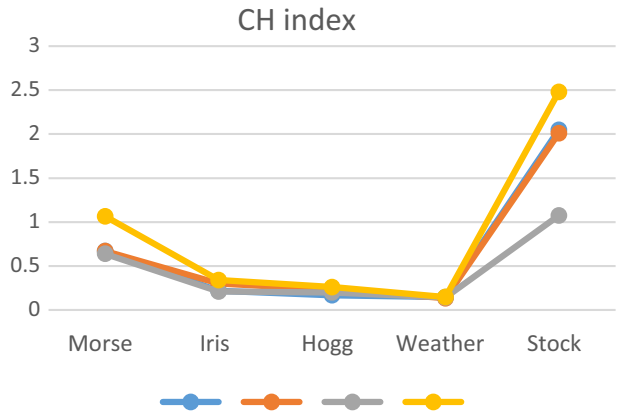

Figure 11. Curve for CH index.

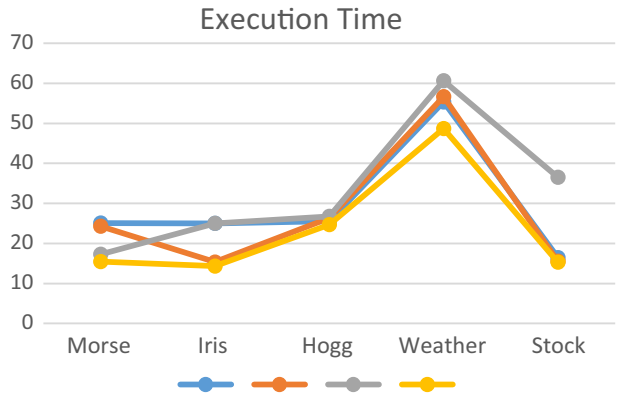

Figure 12. Curve for execution time.

DE approach. The values have been recorded for five different datasets.

9. Conclusion

In this research paper, a variant of the mutation strategy of differential evolution named FSDE is proposed and is applied to k means technique of data clustering. The results obtained show that the variant created is more efficient than the classical schemes of DE and the results are significantly good for clustering application. This method is used for clustering only dataset. Further extension of the work can be done in the field of image and text. Also, FSDE technique is applied only to k means technique of clustering. Further extension of the work can be done in applying FSDE to other techniques of clustering like hierarchical agglomerative method, DBSCAN method etc.

References

Abraham A., Das S., Konar A., Document clustering using differential evolution. In: Evolutionary Computation, CEC 2006, IEEE Congress on IEEE, 2006, pp. 1784–1791.

Alguliev, R.M., Ramiz, M.A., Chingiz, A.M., 2011. Sentence selection for generic document summarization using an adaptive differential evolution algorithm. Swarm Evolut. Comput. 1 (4), 213–222.

Bezdek, J.C., Ehrlich, R., Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. Comput. Geosci. 10 (2), 191–203.

Blake C., Keough E., Merz C.J., UCI Repository of Machine Learning Database, 1998. [Online]. Available: <http://www.ics.uci.edu/~mlearn/MLrepository.html>.

Caliński, T., Harabasz, J., 1974. A dendrite method for cluster analysis. Commun. Stat. Theory Methods 3 (1), 1–27.

Coello, Carlos A., Coello, David A., Van Veldhuizen, Lamont, Gary B., 2002. Evolutionary algorithms for solving multi-objective problems, vol. 242, Kluwer Academic, New York, 2002.

Davies, D.L., Bouldin, D.W., 1979. A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. 2, 224–227.

Fong, S., Suash, D., Xin-She, Yang, Yan, Zhuang, 2014. Towards enhancement of performance of K-means clustering using nature-inspired optimization algorithms. Sci. World J. 2014, 16.

Hatamlou, A., 2013. Black hole: a new heuristic optimization approach for data clustering. Inf. Sci. 222, 175–184.

Indrajit S., Ujjwal M., Nilan J., 2009. Differential fuzzy clustering for categorical data, In: Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on, IEEE, 2009, pp. 1–6.

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

Lloyd, S., 1982. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28, 129–136.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations, In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. 1(14), pp. 281–297.

Maulik, U., Indrajit S., 2010. Automatic fuzzy clustering using modified differential evolution for image classification, Geosci. Remote Sensing, IEEE Trans. 9, 3503–3510.

Maulik, U., Sanghamitra B., Indrajit S., Integrating clustering and supervised learning for categorical data analysis, Syst. Man Cybernet. Part A: Syst. Hum. IEEE Trans. 40(4), 664–675.

Mukherjee, R., Shantanab, D., Das, S., Modified differential evolution with locality induced genetic operators for dynamic optimization, Eur. J.f Oper. Res. 253(2), 337–355.

Paterlini, S., Thiemo K., 2004. High performance clustering with differential evolution. In: Evolutionary Computation, 2004. CEC2004. Congress on, vol. 2. IEEE, pp. 2004–2011.

Pham, D.T., Suarez-Alvarez, M.M., Prostov, Y.I., 2011. Random search with k-prototypes algorithm for clustering mixed datasets, Proc. R. Soc. Lond. A: Math. Phys. Eng. Sci. 467 (2132), 2387–2403. (The Royal Society)

Qu, B.Y., Suganthan, P.N., Jane-Jing, L., 2012. Differential evolution with neighborhood mutation for multimodal optimization. Evolut. Comput. IEEE Trans. 16 (5), 601–614.

Ramadas, M., Pant, M., Abraham, A., Kumar, S., ssFPA/DE: an efficient hybrid differential evolution–flower pollination algorithm based approach, Int. J. Syst. Assurance Eng. Manage. 1–14 (in press).

Saha, S., Bandyopadhyay, S., 2013. A generalized automatic clustering algorithm in a multiobjective framework. Appl. Soft Comput. 13 (1), 89–108.

Singh, V., Saha S., 2014. Modified differential evolution based 0/1 clustering for classification of data points: Using modified new point symmetry based distance and dynamically controlled parameters. In: Contemporary Computing and Informatics (IC3I), 2014 International Conference on, IEEE, pp. 1182–1187.

Storn, R., Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11 (4), 341–359.

Suarez-Alvarez, M.M., Pham, D.T., Prostov, M.Y., Prostov, Y.I., 2012. Statistical approach to normalization of feature vectors and clustering of mixed datasets. In: Proc. R. Soc. A (p. rspa20110704). The Royal Society, 2012.

Thein, Htet Thazin Tike, Khin Mo Mo Tun, 2015. Evaluation of differential evolution and K-means algorithms on medical diagnosis. In: Information Technology: Towards New Smart World (NSITNSW), 2015 5th National Symposium on, IEEE, pp. 1–4.

Wu, G., Rammohan, M., Suganthan, P.N., Wang, R., Huangke, C., 2016. Differential evolution with multi-population based ensemble of mutation strategies. Inf. Sci. 329 (2016), 329–345.

Zaharie, D., 2005. Density based clustering with crowding differential evolution. In: Symbolic and Numeric Algorithms for Scientific Computing, 2005. SYNASC 2005, Seventh International Symposium on, IEEE, p. 8.

Zhang, J., Sanderson, A.C., 2007. An approximate Guassian model of differential evolution with spherical fitness functions. In: Proceedings of the IEEE Congress Evolution Computation, Singapore, pp. 2220–2228.

Zhang, X., Wei, D., Wang, J., Zhongshan, F., Deng, G., 2008. Spatial clustering with obstacles constraints using PSO-DV and K-medoids, In: Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on, vol. 1, IEEE, pp. 246–251.