Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2018) 26: 867 – 881

doi:10.3906/elk-1706-104

Data clustering using eDE, an enhanced differential evolution algorithm with fuzzy c-means technique

Meera RAMADAS[1,*], **Ajith ABRAHAM**[2]
[1]Amity Institute of Information Technology, Amity University, Uttar Pradesh, Noida, India
[2]Machine Intelligence Research Labs (MIR Labs), Auburn, Washington, USA

Received: 09.06.2017 • **Accepted/Published Online:** 29.11.2017 • **Final Version:** 30.03.2018

Abstract: Clustering is the way toward sorting out items into groups whose individuals are comparative somehow. It is a gathering of articles that are intelligent inside, yet unmistakably not at all like the items having a place with different groups. Clustering of data plays a major part in efficient customer segmentation, organization of documents, information retrieval, extraction of topics, classification, collaborative filtering, visualization, and indexing. In the area of information retrieval systems, evolutionary algorithms work in a robust and efficient manner for clustering. To overcome the problem of local maxima, various nature-inspired metaheuristic algorithms like particle swarm optimization, artificial bee colony, and firefly algorithms are considered. In this work, a variant of a differential evolution algorithm named enhanced differential evolution (eDE) is created. eDE is incorporated with the fuzzy c-means technique to perform clustering of data.

Key words: Soft cluster, fuzzy c-means, membership function, validation index

1. Introduction

With improvements in technology, the requirement for attaining, storage, and handling a vast volume of data is ever growing. Clustering is the process of segregating a collection of data into a group of subclasses, called clusters, where the objects within one subclass are more similar to each other than objects within the other subclasses. It is the process of splitting a larger population into smaller groups that are comparable. Distinctions and resemblances are assessed on the basis of attribute properties describing the objects. The similarity measure, its way of implementation, and the method's capability to identify as many hidden patterns possible will define the quality of the clustering results. Organizing the data into clusters shall reflect greater intracluster similarity and less intercluster similarity. In clustering, the characteristics of the data become the key variable of the problem and the choice of their selection within the clustering algorithm will greatly affect the results. Hence, analysis shall always be focused on these characteristics.

Clustering is based upon three characteristics: nesting, exclusiveness, and completeness. In the nested type, separation is built on characteristics of nesting groups. Hierarchical clustering is nested, meaning it gathers data to exist within the bigger clusters. In exclusive separation, the data object is allowed to exist in one or more than one cluster. Completeness is a sort of separation that needs all of the data objects to be collected. In complete clustering, every item is assigned to a group. Several stages are included in data clustering, namely data gathering, initial selection, depiction, clustering tendency, clustering scheme, justification, and analysis.

*Correspondence: meera_mgr@rediffmail.com

Partitional algorithms decompose data into groups of unrelated clusters using a resemblance measure. This is an optimization problem as it diminishes the cluster assigning in terms of probability density. Clustering is used in two manners, namely hard (crisp) and soft (fuzzy). In hard clustering, every element is allotted to one group only. The clusters will be separate but not coinciding. The k-means algorithm is a type of crisp algorithm. In soft clustering, patterns will be allotted to all the groups based on a fuzzy membership pattern. Fuzzy c-means is a type of fuzzy-based clustering. When executing cluster analysis on a dataset, the data are partitioned into groups depending on their similarity. Each member of this group will be assigned a label. Such a group of data is referred to as a cluster. The benefit of performing clustering rather than classification is the flexibility to change and to help differentiate unique features within the group. Clustering is applied in applications such as search engines, web mining, information retrieval, and topological analysis. Since no categorized documents are supplied in clustering, it is also referred to as unsupervised learning. To solve unsupervised learning, various evolutionary algorithms are used. It was proved that evolutionary algorithms work in a robust and efficient manner for clustering according to Coello et al. [1]. These approaches can find global or near global optimal partitions over datasets when the number of clusters is given. Differential evolution (DE) is a simple, stochastic, population-based, easy-to-implement function. It deals with nondifferential, multimodal-based, nonlinear objective functions. Though DE has numerous advantages like ease of use and global exploration, it has a few disadvantages, too. It suffers from premature convergence. Also, the performance of DE decreases when the size of the search space increases. The efficacy and performance of DE is decided by the control parameters and test vector generation strategy. Variants of evolutionary algorithms are created by changing the control parameters to improve the optimization function and also to improve the convergence rate. In this work, we have applied a variant of DE, namely eDE (enhanced differential evolution) to the field of data clustering. In this variant, three control parameters are used: a constant between (0,2), a random variable between (0,1), and the complement of the random variable used. This variant was applied to cluster a standard numerical dataset. eDE gave good results when it was applied for data clustering.

2. Background literature study

Krishna and Murthy [2] suggested a variant of the genetic algorithm (GA) to find a globally ideal partition of a given record into a definite set of clusters. The GA combined with k-means (GKA) uses a distance-based mutation operation. This technique performs faster searches in comparison to other evolutionary algorithms. Bandyopadhyay and Maulik [3] proposed a GA-based k-means clustering that removes the problem of getting stuck in local optimal values during exploitation. This technique was used for clustering the pixels of satellite images of a city in India. Bosman and Thierens [4] discussed the multiobjective evolutionary algorithms and their effects on exploration and exploitation. Hruschka et al. [5] discussed the issues of spontaneously obtaining an optimal partition in bioinformatics datasets. Here, a clustering GA was incorporated in an evolutionary algorithm of clustering (EAC). The EAC arises as a good tool for clustering bioinformatics datasets. Tasoulis [6] proposed a new clustering operator for evolutionary algorithms. It uses the unsupervised k-window clustering algorithm. Results showed that the suggested method is dependable and effective.

Singh and Deb [7] reviewed the works done on multimodal function optimization and provided a detailed examination. A multipurpose hump test function was also introduced. On the basis of comparison, restricted tournament selection and the new test function were shown to be better for finding optimal results. Handl and Knowles [8] developed an evolutionary approach to tackle unsupervised learning problems. The experiment conducted shows the usage of multiobjective clustering in practical performance. Das et al. [9] showed the

application of DE in automatic clustering of large amounts of data. This technique was compared with PSO and the GA. This technique was tested on a real-world dataset. Hruschka et al. [10] did a survey of evolutionary algorithms on clustering. That work mostly dealt with partitional clustering that looks for hard clusters.

Das and Sil [11] proposed a reformed DE algorithm for bundling pixels of images in gray scale. This technique uses a kernel-induced similarity measure making partitioned data linearly nonseparable. This method was tried on ten gray-scale images and the new technique proved to be efficient. Karaboga and Ozturk [12] used the artificial bee colony (ABC) for clustering data on benchmark problems. This was compared with results from PSO and other evolutionary techniques. The results obtained show the efficiency of the method. Chen and Ye [13] used a particle swarm-based clustering called PSO clustering. PSO is used to search for cluster centers. It is a simple technique and its efficiency was tested on four artificial datasets. Hatamlou [14] developed a new heuristic algorithm on the basis of black-hole phenomena. The black-hole algorithm developed was tested for clustering data and the performance was justified. Xu et al. [15] introduced an evolutionary clustering concept by tracing the time-varying proximities between objects. This technique was used on numerous static clustering algorithms. Results showed that the new method outdoes the existing techniques. Ozturk et al. [16] proposed new solution mechanism with discrete ABC. The performance superiority was shown by comparing the basic evolutionary approaches of ABC, PSO, and GAs. Ramadas et al. [17] proposed a variant using DE with a flower pollination algorithm. This technique was tested for efficiency and it was applied to clustering of five numeric datasets. The technique used was proved to be efficient in comparison to other evolutionary concepts.

3. Fuzzy c-means clustering

In a few cases of clustering, the clusters formed are not well separated. In fuzzy set theory, objects fit in a cluster with a membership degree between 0 and 1. Fuzzy c- means (FCM) is related to the k-means technique. FCM was developed by Dunn [18]. In FCM, the dataset is clustered into n groups where each datum in the dataset belongs to a particular cluster with a certain degree. It is a type of soft clustering where a datum can belong to multiple clusters. Each datum has a membership grade, which indicates the degree to which data belong to a particular cluster. A point on the edge of the cluster will have a lower membership degree compared to other data in the cluster. Assume a set of n objects as $x_i = \{x_1, x_2, \ldots .x_n\}$. Here, a collection of k clusters are initialized as $C_1C_2, \ldots C_k$ and a partition matrix $W = w_{ij} \in [0,1]$ for $i = 1, 2...n$ and $j = 1, 2...k$. w_{ij} denotes the membership degree of the objects.

For a cluster C_j , the centroid c_j is denoted as:

$$c_j = \frac{\sum_{i=1}^{n} w_{ij}^p x_i}{\sum_{i=1}^{n} w_{ij}^p}, \tag{1}$$

where p is the fuzzifier, which denotes the level of fuzziness of the cluster. Larger $p$means a lower membership degree. The membership degree is computed as:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{1}{p-1}}}. \tag{2}$$

4. Classical differential evolution

In the n-dimensional search space, a designated number of vectors are identified casually. In every iteration two or more vectors are chosen arbitrarily from the population and are combined to form a new vector. A trial vector is created by equating the resultant vector with the predecided target vectors. If trial vector provides a fitter objective function, then the trial vector is acknowledged into the next generation. Mutation, recombination, and selection are pursued until some stopping criterion is attained. DE utilizes the population of NP candidate solutions indicated by $X_{i,G}$, where index $i = 1, 2...NP$ constitutes the population while G represents the generation of the population.

Mutation: The weighted difference of vectors in the generation is calculated. For any given variable $X_{i,G}$, arbitrarily choose three vectors $X_{r1,G}, X_{r2,G}$, and $X_{r3,G}$ where r_1, r_2, r_3 are dissimilar from each other. Subsequently, donor vector $V_{i,G}$ is calculated as:

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

where mutation factor F is a constant from (0, 2). The above strategy is denoted as DE/rand/1. The mutation function distinguishes one DE strategy from the other. The other main mutation strategies used are given below:

$$\text{DE/rand/2 } V_{i,G} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G}) + F.(X_{r4,G} - X_{r5,G}), \tag{4}$$

$$\text{DE/best/1 } V_{i,G} = X_{best,G} + F.(X_{r1,G} - X_{r2,G}), \tag{5}$$

$$\text{DE/best/2 } V_{i,G} = X_{best,G} + F.(X_{r1,G} - X_{r2,G}) + F.(X_{r3,G} - X_{r4,G}), \tag{6}$$

$$\text{DE/rand-to-best/1 } V_{i,G} = X_{r1,G} + F.(X_{best,G} - X_{r2,G}) + F.(X_{r3,G} - X_{r4,G}). \tag{7}$$

Crossover/recombination: This operation uses prosperous solutions in the population. Trial vector $U_{i,G}$ is generated for target vector $X_{i,G}$ by means of binomial crossover. Using probability $C_r \in [0, 1]$, the elements of the donor vector go into the trial vector. Crossover probability C_r is designated along with population size NP.

$$U_{j,i,G+1} = \begin{cases} V_{j,i,G+1} \text{ if } rand_{i,j}[0,1] \leq C_r \; or if j = I_{rand} \\ X_{j,i,G+1} \text{ if } rand_{i,j}[0,1] > C_r \text{ or if } j \neq I_{rand} \end{cases} \tag{8}$$

Here $rand_{i,j} \approx \cup[0, 1]$ and I_{rand} is a random numeral from 1,2...N.

Selection: Target vector $X_{i,G}$ shall be coupled to the trial vector $V_{i,G}$, taking the lowest resultant of the function to the subsequent generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} \text{ if } f(U_{i,G+1}) \leq f(X_{i,G}) \text{ where } i = 1, 2, ...N \\ X_{i,G} \text{ otherwise} \end{cases} \tag{9}$$

5. Enhanced differential evolution

A new strategy has been proposed for mutation called eDE. This strategy uses three control parameters. Parameter F takes a constant value between (0,2). Parameter *F1* takes a varying value that lies between (0,1) and *F2* takes the complement of *F1*. As F1 and F2 have random values in each iteration, the convergence

behavior is enhanced profoundly. As three different control parameters are considered, the value of the donor vector is enhanced significantly and hence the effectiveness of the eDE algorithm is heightened profoundly. Here, two sets of difference vectors are used so the preferred perturbation is attained faster. By randomly choosing the vectors, the algorithm is prevented from being greedy. The mutation strategy for the proposed technique is given as:

$$X^{'} = F \times (X_{r1,G}) + F1 \times (X_{best,G} - X_{r2,G}) - F2 \times (X_{best,G} - X_{r3,G}). \tag{10}$$

By utilizing the best vector value, the algorithm converges faster in comparison to the traditional strategies. Using more than one difference vector increases the diversity of the population considered. The flow diagram for the proposed technique is shown in Figure 1. The crossover and selection for eDE is same as in the classical DE technique.

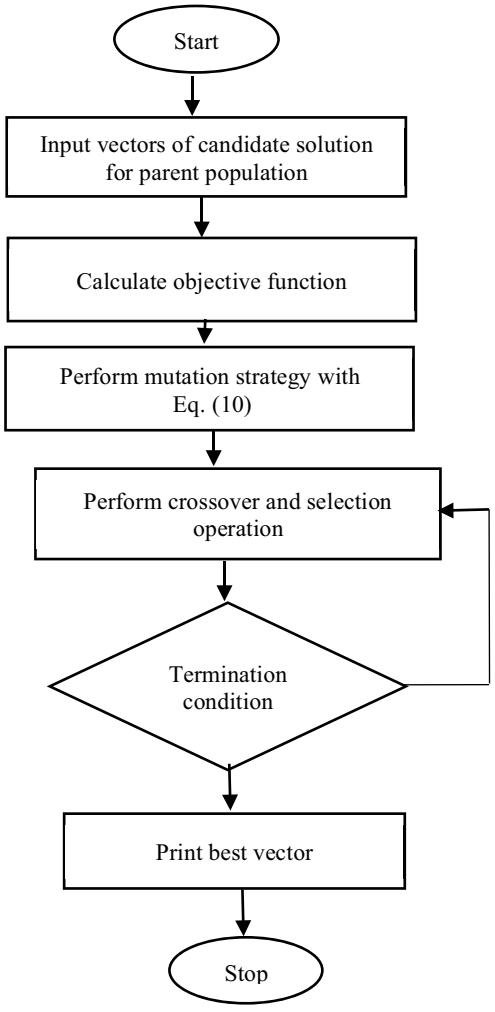

Figure 1. Flow chart for eDE.

6. Experimental setting

eDE was executed using MATLAB r2008b and a relative analysis was acquired with five diverse mutation strategies of the classical DE algorithm and eDE. Fifteen diverse functions were considered and the results were

computed. The optimal value of control parameter F was set as 0.6. As larger values of CR speed up the convergence rate, for this experiment, CR was set to 0.8. The value to reach (VTR) is the global minimum or maximum of the function to stop the optimization if it is reached. The results are formulated for comparison with the existing algorithms in Table 1. By fixing the dimension as 50 and the VTR as e-015, the best value and the CPU time of diverse function strategies were considered. The best values for each function are indicated in bold in Table 1. The experiment substantiated that eDE produced better results in the case of numerous standard functions. The experiment was also validated by changing the dimension and VTR value.

Table 1. Best values obtained after 25 runs for diverse functions (in bold).

Function	DE					
	DE/best/1	DE/rand/1	DE/rand-to-best/1	DE/best/2	DE/rand/2	eDE
Sphere	9.73×10^{-16}	6.90×10^{-16}	7.53×10^{-16}	9.66×10^{-16}	7.17×10^{-16}	$\mathbf{6.04\times10^{-16}}$
Beale	3.27×10^{-16}	$\mathbf{2.32\times10^{-16}}$	3.71×10^{-16}	7.59×10^{-16}	7.73×10^{-16}	3.95×10^{-16}
Booth	3.50×10^{-16}	2.05×10^{-16}	6.07×10^{-16}	7.08×10^{-16}	8.35×10^{-16}	$\mathbf{1.36\times10^{-16}}$
Schwefel	$\mathbf{-1.80\times10^{3}}$	$\mathbf{-2.25\times10^{3}}$	$\mathbf{-7.84\times10^{1}}$	$\mathbf{-1.38\times10^{3}}$	$\mathbf{-1.66\times10^{3}}$	$\mathbf{-2.10\times10^{3}}$
Michlewicz	**−7.64**	**−7.21**	**−7.39**	**−6.95**	**−6.84**	**−6.6**
Schaffer N.2	6.60×10^{-16}	8.88×10^{-16}	4.43×10^{-16}	6.55×10^{-16}	8.87×10^{-16}	$\mathbf{2.22\times10^{-16}}$
Schaffer N.4	3.05×10^{-15}	2.90×10^{-1}	2.92×10^{-1}	2.93×10^{-1}	2.89×10^{-1}	$\mathbf{2.82\times10^{-1}}$
HimmelBlau	1.60×10^{-16}	8.05×10^{-16}	3.83×10^{-16}	9.12×10^{-16}	$\mathbf{1.46\times10^{-16}}$	3.35×10^{-16}
Bird	$\mathbf{-1.04\times10^{-02}}$	$\mathbf{-1.07\times10^{-2}}$	$\mathbf{-1.05\times10^{-2}}$	$\mathbf{-1.07\times10^{-2}}$	$\mathbf{-1.03\times10^{-2}}$	$\mathbf{-1.03\times10^{-2}}$
Extended Cube	3.31×10^{-15}	$\mathbf{4.98\times10^{-5}}$	6.10×10^{-8}	1.93×10^{-5}	2.68	8.60×10^{-15}
Ackeley	7.19×10^{-15}	$\mathbf{6.46\times10^{-12}}$	7.99×10^{-15}	3.63×10^{-13}	3.09	1.50×10^{-14}
Gold	3.00	3.00	3.00	3.00	3.00	3.00
Griewank	9.99×10^{-16}	9.99×10^{-16}	1.60×10^{-13}	6.56×10^{-13}	1.07	$\mathbf{2.40\times10^{-12}}$
Rastrigin	1.79×10^{1}	1.23×10^{2}	7.47×10^{1}	1.28×10^{2}	1.52×10^{2}	2.98×10^{1}
Rosenbrock	9.60×10^{-16}	$\mathbf{1.07\times10^{-8}}$	7.88×10^{-16}	3.90×10^{-9}	1.07×10^{1}	1.50×10^{-8}

7. Statistical analysis

The Friedman test was implemented on the results given in Table 1, and the results attained are formulated in Table 2. The statistical analysis verifies the efficiency of the eDE algorithm. Table 3 shows the rank of various mutation strategies used based on the best value and CPU time. The tables show that eDE has significantly better performance in comparison to the existing mutation strategies. The rank obtained on the basis of CPU time is the best for eDE and the rank attained on the basis of best value is better for eDE in comparison to the traditional mutation strategies considered. These rankings obtained on the basis of the Freidman test validate the efficiency of the eDE technique. The rank obtained on the basis of CPU time taken is depicted in Figure 2.

Table 2. Test statistics using the Friedman test.

N	50
Chi-square	14.54
Degrees of freedom	5
Asymptotic significance	0.002

Table 3. Ranks of the different strategies.

Strategies	Mean rank on best value	Mean rank on CPU time
DE/best/1	3.4	4.7
DE/rand/1	3.7	3.33
DE/best-to-rand/1	2.7	3.67
DE/best/2	4.06	3.7
DE/rand/2	3.9	4.1
eDE	3.1	1.6

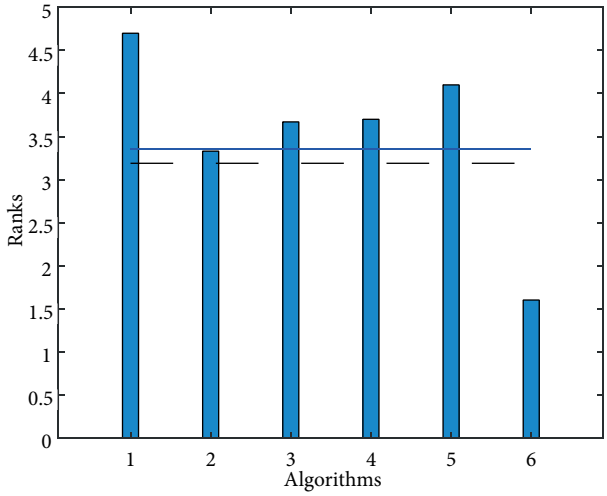

Figure 2. Bonferroni–Dunn bar chart for ranks based on CPU time for various strategies.

8. eDE in FCM clustering

The eDE algorithm is used in clustering using the FCM technique. Fitness of every variable is accomplished by estimating the distance between the centroid and the entity point. It is represented as:

$$Fitness(C) = \sum_{j=1}^{k} \sum_{i=1}^{n} w_{ij} \left\| x_i^j - c^j \right\|^2, \tag{11}$$

where x_i^j is the entity point, c^j is the centroid, w_{ij} is the membership degree, and $\left\| x_i^j - c^j \right\|$ provides the distance between the centroid and the entity point. The flow chart for the clustering technique using the variant of DE is given in Figure 3.

9. Experimental results on clustering

The experiment was run on six standard datasets with numeric data to compare the execution of the k-means algorithm, GA, PSO, and classical DE with eDE in clustering. The FCM technique of clustering was incorporated with the GA, PSO, classical DE, and eDE for execution of data clustering. The resultant cluster graph and curve graph for each dataset were attained. The cluster qualities of the clusters acquired were compared. Eight real-time datasets from the MATLAB repository were used. The datasets used are described below:

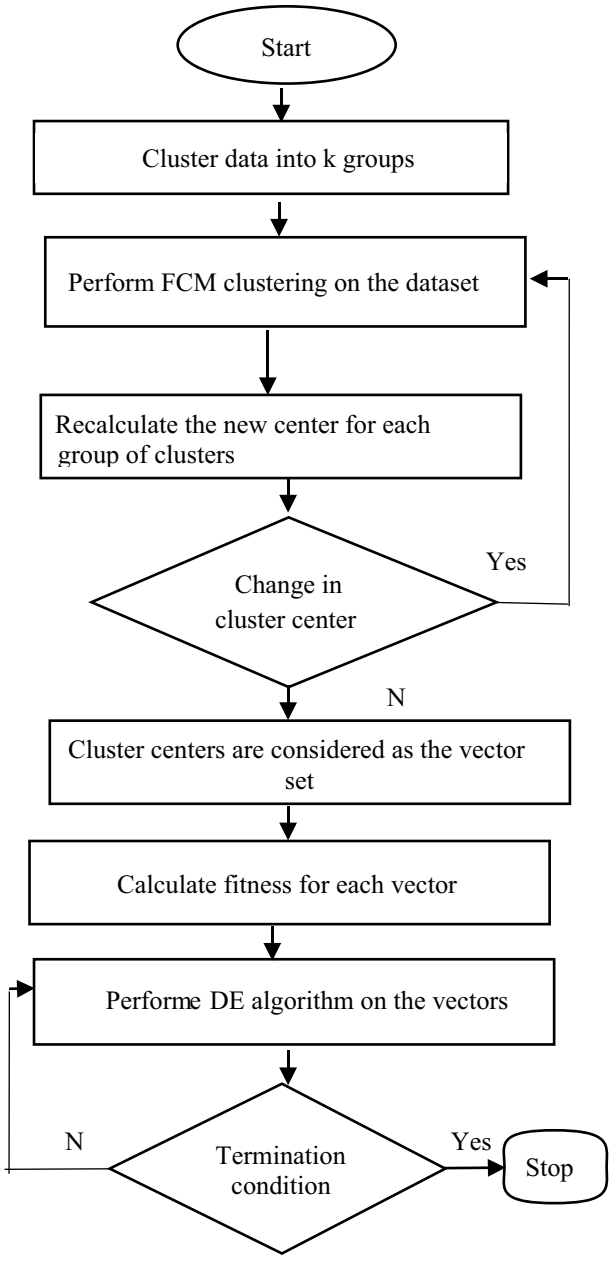

Figure 3. Flow chart for eDE in FCM clustering.

- Moore dataset (n = 120, d = 6, k = 2): This dataset is a lab-based report of the biochemical demand with five predictors.
- Cities (n = 2961, d = 9, k = 2): This dataset gives the quality of life ratings for US metropolitan areas.
- Kmeansdata (n = 100, d = 4, k = 3): It is a four-dimensional data provided for clustering.
- Reactions (n = 75, d = 3, k = 3): It is a reaction kinetic that is a function of three chemical substances: hydrogen, isopentane, and n-pentane. It uses the Hougen–Watson model.

- 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.
- Examgrades (n = 649, d = 33, k = 3): The dataset represents student performance from two Portuguese schools. Grades of students and demographic and social features of the school were gathered from school reports and a standard questionnaire and used as data attributes. Performances of students in two distinct subjects, namely mathematics (mat) and Portuguese language (por), were given as two datasets.
- Topography (n = 64,800, d = 256, k = 3): The dataset contains numerous representations of the earth's topography. The data were supplied by the National Geographical Data Center, NOAA, US Department of Commerce, under data announcement 88-MGG-02.
- Gatlin (n = 307,200, d = 640, k = 3): The dataset contains numeric data with 640 different attributes.

These datasets were used as input for clustering and results were acquired for different algorithms under consideration. The cluster graph and curve graph for the kmeansdata dataset are given in Figure 4. In the cluster graph, the x-axis shows the position and the y-axis shows the distance of data of the kmeansdata set. The curve graphs obtained during various iterations of clustering using the DE and eDE algorithms with k-means for the kmeansdata dataset is depicted where the x-axis shows the number of iterations and the y-axis shows the best cost obtained at each iteration.

9.1. Validation indexes

There are various quantitative evaluation techniques available to test the cluster quality and these are known as validation indexes. Numerous validation indexes are used for testing the quality of clusters obtained using the FCM technique. It is used as a tool by researchers to test the cluster result. The following are the various validation indexes considered.

9.1.1. Partition coefficient (PC)

The PC defines the extent of overlapping between clusters [19]. Higher values of PC give good clusters. The formula for the PC is given as:

$$PC = \frac{1}{N}\sum_{i=1}^{c}\sum_{j-1}^{N}\left(\mu_{ij}\right)^2, \tag{12}$$

where μ_{ij} is the membership of data j in cluster i . The comparative results for the PC are given in Table 4. The result obtained for eDE is best in comparison to the other techniques.

9.1.2. Classification entropy (CE)

CE calculates the fuzziness of the cluster partition. The formula is given as:

$$CE = -\frac{1}{N}\sum_{i=1}^{c}\sum_{j-1}^{N}\mu_{ij}\log\left(\mu_{ij}\right). \tag{13}$$

The higher the value of CE, the better the clusters will be. The comparative results obtained for CE for the various algorithms are given in Table 5. The result obtained for eDE is best in comparison to the other techniques.

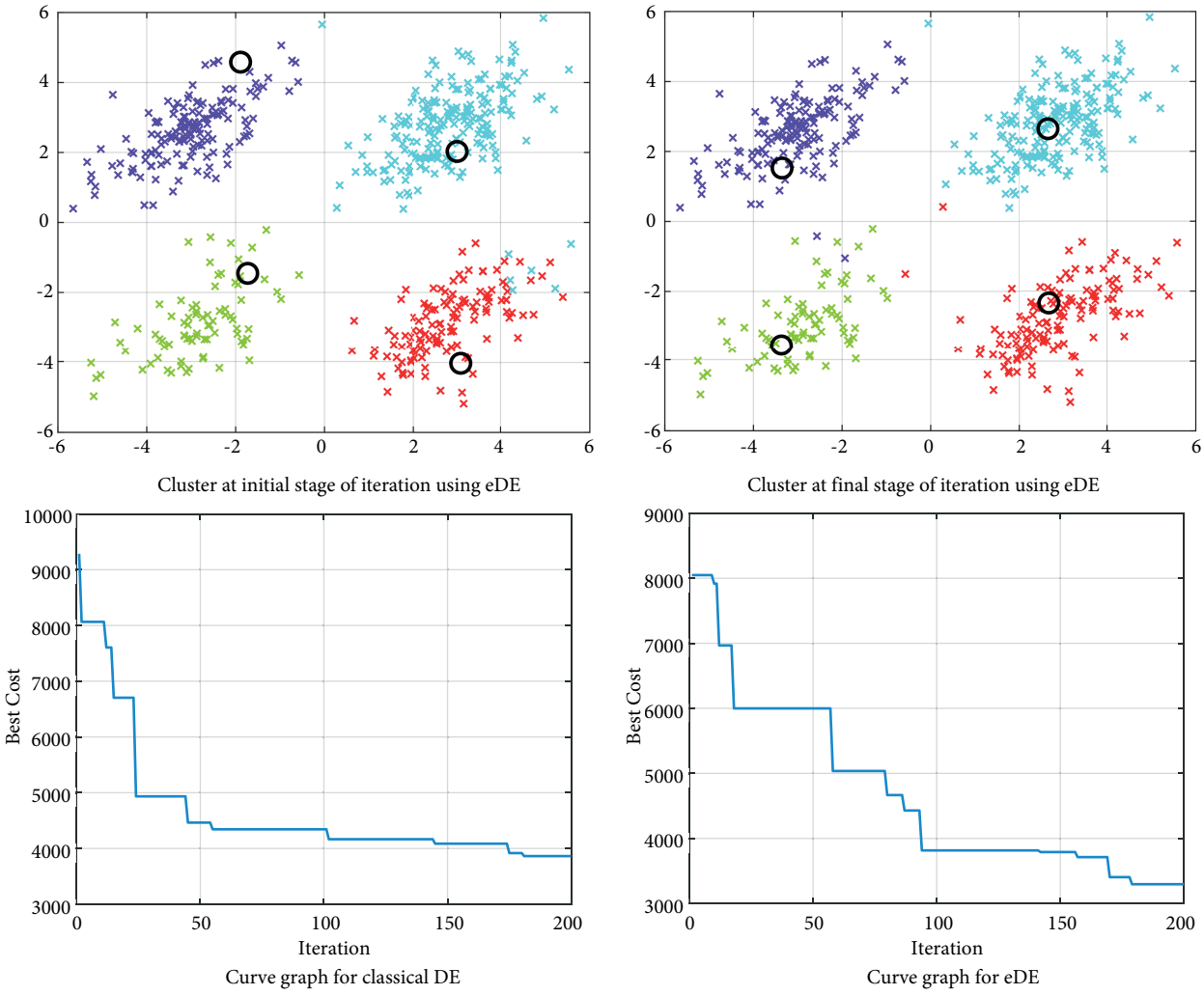

Figure 4. Cluster diagram and cluster graph of kmeansdata dataset.

Table 4. Comparative table for PC.

Datasets	Partition coefficient				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.765	0.767	0.761	0.764	0.7677
Iris	0.747	0.723	0.712	0.771	0.773
Examgrades	0.511	0.511	0.482	0.511	0.5112
Moore	0.656	0.657	0.654	0.661	0.6615
Cities	0.366	0.355	0.344	0.313	0.3666
Reactions	0.851	0.813	0.822	0.844	0.8524
Topography	0.534	0.507	0.511	0.531	0.545
Gatlin	0.572	0.54	0. 516	0.5624	0.581

Table 5. Comparative table for CE.

Datasets	Classification entropy				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.445	0.443	0.446	0.447	0.4474
Iris	0.467	0.477	0.473	0.477	0.489
Examgrades	0.832	0.812	0.824	0.837	0.84
Moore	0.612	0.622	0.625	0.621	0.633
Cities	1.473	1.412	1.474	1.472	1.477
Reactions	0.775	0.786	0.772	0.781	0.781
Topography	0.75	0.698	0.802	0.8113	0.82
Gatlin	0.65	0.71	0.69	0.747	0.748

9.1.3. Partition index (SC)

SC defines the ratio of the sum of compactness to separation of the clusters. The formula for SC is given as:

$$SC = \sum_{i=1}^{c} \frac{\sum_{j=1}^{N} (\mu_{ij})^m \left\| x_j - v_i \right\|^2}{N_i \sum_{k=1}^{c} \left\| x_k - v_j \right\|^2}, \tag{14}$$

where v_i is the cluster center of cluster i, N_i is the number of objects in cluster i,m is the weighting exponent, and $\left\| x_j - v_i \right\|^2$ is the distance between particles and centroid. The lower the value of SC, the better the formed cluster is. The results obtained for SC are tabulated in Table 6. The result obtained for eDE is best in comparison to the other techniques.

Table 6. Comparative table for SC.

Datasets	Partition index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.707	0.712	0.699	0.682	0.665
Iris	0.623	0.633	0.612	0.615	0.611
Examgrades	2.012	2.12	2.02	1.89	1.88
Moore	0.7123	0.72	0.689	0.677	0.665
Cities	12.23	12.1	12.18	12.13	12.12
Reactions	2.627	2.58	2.63	2.54	2.54
Topography	0.168	0.168	0.166	0.1646	0.1644
Gatlin	0.079	0.0792	0.078	0.0809	0.08

9.1.4. Separation index (S)

This index deals with the minimum distance separation for cluster validity. The formula for separation index is given as:

$$S = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^2 \|x_j - v_i\|^2}{N \min_{i,k} \|v_k - v_i\|^2}. \quad (15)$$

The lower the separation index, the better the formed cluster is. The results obtained for the separation index are tabulated in Table 7. The result obtained for eDE is best in comparison to the other techniques.

Table 7. Comparative table for separation index.

Datasets	Separation index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.0015	0.0012	0.002	0.001	0.0008
Iris	0.0063	0.0068	0.006	0.005	0.0054
Examgrades	0.0285	0.0292	0.291	0.286	0.2852
Moore	0.0572	0.0587	0.058	0.057	0.0571
Cities	0.0915	0.0923	0.092	0.093	0.0898
Reactions	0.087	0.089	0.092	0.086	0.082
Topography	0.002	0.0021	0.0013	0.0012	0.001
Gatlin	2.18e-04	2.17e-04	2.2e-04	2.14e-04	2.14e-04

9.1.5. Xie Beni Index (XB)

The XB index was proposed by Xie and Beni [20]. It measures the ratio of total variation within clusters to separation of clusters. XB is given as:

$$XB = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(\mu_{ij})^m \|x_j - v_i\|^2}{N \min_{i,j} \|x_j - v_i\|^2}. \quad (16)$$

The lower the XB value, the better the formed cluster is. The results obtained for XB are given in Table 8. result obtained for eDE is best in comparison to the other techniques.

Table 8. Comparative table for XB.

Datasets	Xie Beni index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	4.987	4.97	4.82	4.78	4.78
Iris	3.804	3.812	3.807	3.803	3.801
Examgrades	1.267	1.278	1.264	1.212	1.189
Moore	1.431	1.442	1.414	1.387	1.36
Cities	1.473	1.456	1.453	1.452	1.401
Reactions	28.53	28..12	28.78	27.89	27.76
Topography	0.821	0.821	0.818	0.8053	0.812
Gatlin	0.975	0.966	0.975	0.9623	0.9601

9.1.6. Execution time

It is the total time taken for the execution of a task. The lower the execution time, the better the cluster is. Execution times for the various algorithms are shown in Table 9. The result obtained for eDE is best in comparison to the other techniques.

Table 9. Comparative table for execution time.

Datasets	Execution time				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	13.45	13.12	13.54	13.12	13.01
Iris	15.1	25.018	15.34	25.03	14.13
Examgrades	11.3	10.67	12.3	11.23	11.21
Moore	8.14	8.87	8.34	9.99	8.12
Cities	37.1	37.6	36.7	37.24	36.9
Reactions	15.43	16.12	15.67	17.23	15.12
Topography	0.813	0.8002	0.813	0.8053	0.8021
Gatlin	0.943	0.975	0.954	0.9623	0.9613

9.1.7. Dunn index (DI)

It is a matrix for evaluating the cluster quality. It is a function of the ratio of the sum of intradistances to interdistances [19]. It tries to find a good intracluster and intercluster association. It is the ratio of the intercluster to intracluster distance of the clusters. Here, the larger the value of the DI index, the better the formed clusters are. The formula for the DI index is given as:

$$DI = \min\left\{\frac{\min d\left((c_i, c_j)\right)}{\max d`(l)}\right\}, \tag{17}$$

where i and j are cluster labels, $d`(k)$is the average distance between cluster elements to the center of cluster l, and $d\left((c_i, c_j)\right)$ is the distance between these centroids. The comparative results obtained for the DI index for the various algorithms are shown in Table 10. The result obtained for eDE is best in comparison to the other techniques.

Table 10. Comparative table for DI index.

Datasets	Dunn index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.0222	0.018	0.02	0.011	0.0255
Iris	0.028	0.03	0.03	0.034	0.034
Examgrades	0.128	0.11	0.114	0.1189	0.1289
Moore	0.221	0.22	0.234	0.255	0.255
Cities	0.045	0.04	0.067	0.056	0.0691
Reactions	0.443	0.33	0.333	0.441	0.443
Topography	0.131	0.134	0.121	0.135	0.139
Gatlin	0.087	0.0767	0.0862	0.0905	0.0909

9.2. Alternative Dunn index (ADI)

It is a variation of the DI index. The larger the value of the ADI index, the better the cluster that is formed. The formula for the ADI index is given as:

$$ADI = \min\left\{\frac{d\left(x_i, v_i\right) - d(x_j, v_j)}{\max d`(l)}\right\}, \tag{18}$$

where v_i shows the center of cluster i. The relative comparison obtained for the ADI index is given in Table 11. The result obtained for eDE is best in comparison to the other techniques.

Table 11. Comparative table for ADI index.

Datasets	ADI index				
	K means	GA	PSO	Classical DE	eDE
Kmeansdata	0.001	0.0013	0.002	0.001	0.0021
Iris	0.008	0.0068	0.009	0.009	0.0096
Examgrades	0.051	0.045	0.034	0.048	0.0531
Moore	0.045	0.036	0.043	0.046	0.0489
Cities	0.001	0.0010	0.001	0.002	0.0029
Reactions	0.114	0.113	0.111	0.013	0.115
Topography	9.23e-04	9.43e-04	9.28e-04	9.5e-04	9.5e-04
Gatlin	5.18e-05	5.23e-05	5.16e-05	5.15e-05	5.17e-05

9.3. Graphical representation

The tabulated values of the validation index have been depicted graphically. Figure 5 shows the performance curve for the DI validation index. The x-axis represents the different datasets used and the y-axis represents the value obtained. The line graph shows that the values obtained for the eDE are better than the values obtained from the classical DE approach. The values have been recorded for eight different datasets.

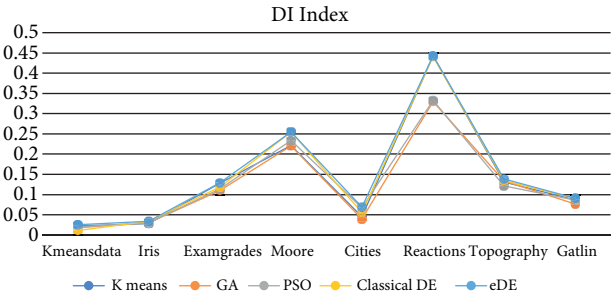

Figure 5. Curve for DI index.

10. Conclusion

In this work, a new variant of DE was proposed and named eDE. This was compared with diverse mutation strategies of DE. The comparative study showed the superior performance of eDE. The ranks computed also justified the efficiency of the strategy. eDE was applied with the FCM technique for clustering standard numeric

datasets. The cluster quality showed the efficiency of the variant developed. This work can be extended to the fields of image thresholding, texture enhancement, etc. for displaying the performance of the new mutation strategy in those areas.

References

[1] Coello CAC, Veldhuizen VDA, Lamont GB. Evolutionary Algorithms for Solving Multi-Objective Problems (Vol. 242). New York, NY, USA: Kluwer Academic, 2002.

[2] Krishna K, Murty NM. Genetic K-means algorithm. IEEE T Syst Man Cy B 1999; 29: 433-439.

[3] Bandyopadhyay S, Ujjwal M. An evolutionary technique based on K-means algorithm for optimal clustering in RN. Inform Sciences 2002; 146: 221-237.

[4] Bosman PAN, Thierens D. The balance between proximity and diversity in multi-objective evolutionary algorithms. IEEE T Evolut Comput 2003; 7: 174-188.

[5] Hruschka ER, de Castro LN, Campello RJGB. Evolutionary algorithms for clustering gene-expression data. In: IEEE Data Mining; 1–4 November 2004; Brighton, UK. New York, NY, USA: IEEE. pp. 403-406.

[6] Tasoulis DK, Plagianakos VP, Vrahatis MN. Clustering in evolutionary algorithms to efficiently compute simultaneously local and global minima. In: IEEE 2005 Evolutionary Computation; 2–5 September 2005; Edinburgh, UK. New York, NY, USA: IEEE. pp. 1847-1854.

[7] Singh G, Deb K. Comparison of multi-modal optimization algorithms based on evolutionary algorithms. In: ACM Genetic and Evolutionary Computation Conference; 8–12 July 2006; Seattle, WA, USA. New York, NY, USA: ACM. pp. 1305-1312.

[8] Handl J, Knowles J. An evolutionary approach to multiobjective clustering. IEEE T Evolut Comput 2007; 11: 56-76.

[9] Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm. IEEE T Syst Man Cy A 2008; 38: 218-237.

[10] Hruschka ER, Campello RJGB, Freitas AA, de Carvalho ACPLF. A survey of evolutionary algorithms for clustering. IEEE T Syst Man Cy C 2009; 39: 133-155

[11] Das S, Sil S. Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm. Inform Sciences 2010; 180: 1237-1256.

[12] Karaboga D, Ozturk C. A novel clustering approach: artificial bee colony (ABC) algorithm. Appl Soft Comput 2011; 11: 652-657.

[13] Chen CY, Ye F. Particle swarm optimization algorithm and its application to clustering analysis. In: IEEE Electrical Power Distribution Networks; 2–3 May 2012; Tehran, Iran. New York, NY, USA: IEEE. pp. 789-794.

[14] Hatamlou A. Black hole: a new heuristic optimization approach for data clustering. Inform Sciences 2013; 222: 175-184.

[15] Xu KS, Kliger M, Hero AO 3rd. Adaptive evolutionary clustering. Data Min Knowl Disc 2014; 28: 304-336.

[16] Ozturk C, Hancer E, Karaboga D. Dynamic clustering with improved binary artificial bee colony algorithm. Appl Soft Comput 2015; 28: 69-80.

[17] Ramadas M, Abraham A, Kumar S. Using data clustering on ssFPA/DE-a search strategy flower pollination algorithm with differential evolution. In: Springer Hybrid Intelligent Systems; 21–23 November 2016; Marrakech, Morocco. Berlin, Germany: Springer. pp. 539-550.

[18] Dunn JC. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J Cybernetics 1973; 3: 32-57.

[19] Bezdek JC. Cluster validity with fuzzy sets. J Cybernetics 1973; 3: 58-73.

[20] Xie XL, Beni G. A validity measure for fuzzy clustering. IEEE T Pattern Anal 1991; 13: 841-847.