Research Journal of Applied Sciences 9 (7): 452-460, 2014
ISSN: 1815-932X

Differential Evolution for Fuzzy Clustering Using Self-Adaptive Trade-Off Between Exploitation and Exploration

[1]Siriporn Supratid and [2]Phichete Julrode
[1]Faculty of Information Technology, Rangsit University, Pathum-Thani, Thailand
[2]Faculty of Science and Technology, Phuket Rajabhat University, Phuket, Thailand

Abstract: Differential Evolution (DE) has emerged as one of the fast and efficient search heuristics of current interest. Combining DE and Fuzzy C-Means (DEFCM) explicitly improves the clustering on the basis of degree of membership. However, misdirection of the search, e.g., too much either exploitation or exploration search still ruin the achievement of global optimal solution. Thereby, this study proposes a DE-based fuzzy clustering using self-adaptive trade-off between exploitation and exploration (DEFSA). The efficiently dynamic trade-off is controlled by none of arbitrarily defined parameters. The performance measurements relate to F-measures, FCM objective degree and Xie-Beni validity index. The experiments are operated on real-world as well as artificial data sets. The results show the superior performance of the proposed method in terms of clustering correctness over traditional fuzzy ant-based clustering as well as some other efficient clustering methods.

Key words: Differential evolution, fuzzy C-means, non-parametric, exploration, exploitation

INTRODUCTION

Clustering is one of the most important unsupervised learning techniques (Rojas, 1996; Herrero *et al*., 2011). It organizes a set of sample cases into similar groups called clusters. The objects within one cluster are highly similar and dissimilar with others in different clusters. Clustering is widely applied in several application fields such as pattern recognition (Webb, 2002), data mining (Tan *et al*., 2004), machine learning (Alpaydın, 2004), etc. For solving clustering problems, efficient approaches such as Self-Organizing feature Maps (SOM) (Kohonen, 1995), Average Linkages (AL) (Hastie *et al*., 2009) have been successfully applied. Fuzzy C-Means (FCM) (Dunn, 1973; Bezdek, 1981), a partitional type of soft clustering employs basic idea relating to find cluster centers, then refining them such an algorithm allows each sample cases belonging to two or more clusters with different degrees of membership. Thus, FCM is well applied to real-world applications. Nevertheless, FCM is sensitive to initialization and can be easily trapped into local optimal solutions. In order to relieve such difficulty, most of the researches are proposed, aiming to the integration between FCM and powerful evolutionary optimization algorithms, e.g., Particle Swarm Optimization (PSO) (Tin *et al*., 2012; Izakian and Abraham, 2011) as well as Ant Colony Optimization (ACO) (Dogan and Korurek, 2012). Differential Evolution (DE) a Heuristic algorithm was put forward by Storn and Price (1997). DE exhibits remarkable performance in optimizing a wide variety of optimization problems in terms of final accuracy, convergence speed and robustness. The runtime complexity and the function evaluation number for acquiring global minimizer by DE algorithm is generally smaller than the compared algorithms including PSO (Civicioglu and Besdok, 2013). In addition, it is found by Montgomery *et al.* (2011) that DE can deliver results of comparable quality significantly faster than ACO for RFID antenna design. One of the reasons is that ACO evolution is based on the pheromone levels left by the ants and hence does not contain such a direct link to knowledge contained in previous iterations as DE. By such competent characteristics, several number of DE variants are utilized to solve several optimization problems and generates superior outcomes than other conventional evolutionary approaches. A new DE mutation strategy, Sayah *et al.* (2013) shows the improvement of the performance and the alleviation of premature convergence in economic dispatch problem. Another mutation strategies, proposed by Wang and Zhao (2013) dynamically control parameters in self-adaptive manner. The research (Wang *et al*., 2013) adopts different DE mutation operators of for each sub-population the better results over the past DE variants successfully are shown. DE relieves the problem regarding a sensitivity of FCM to the initial state the aim is to solve machine cell formation problems in a fuzzy environment (Kao and Chen, 2013).

Corresponding Author: Siriporn Supratid, Faculty of Information Technology, Rangsit University, 12000 Pathum-Thani, Thailand

Moreover, achievement of DE-based fuzzy clustering on some benchmark data sets is shown by Ravi *et al*. (2010). Likewise, the fuzzy differential evolution (Vucetic and Simonovic, 2013) is successfully employed to solve a reservoir operation problem fast convergence is achieved as well. A chaotic differential evolution algorithm is proposed by Li *et al*. (2013) competition co-evolution among each sub-population is pursued to improve the performance of the conventional DE.

However, any optimization algorithm including DE which exploits the search space around the optimally best solution may not get globally best solution. Vice versa, the one which explores the search space possibly gets better solution by enhancing the diversity of solutions but needs more time to converge. The method, proposed by Slowik (2011) applies fuzzy control on trade-off between exploration and exploitation of evolutionary algorithm search a number of generations of evolutionary algorithm and population diversity are considered in such control. Han *et al*. (2013) proposes a modified DE mutation operation considers local information nearby each individual population to trade-off between the exploration ability and the exploitation ability. These two latter works, represent examples of the attempt to achieve globally best solution by striking the trade-off between exploration and exploitation. Those types of research usually proceed such trade-off by using particular control parameters. This may lead to a biased and overly optimistic clustering process thus limit the usefulness of the model. There exist some classification and regression works using non-controlling parameters technique (Aydin, 2007; Pai *et al*., 2012; Memmedli and Nizamitdinov, 2012). They considerably improves the performance accuracy as well as reliability.

This study, thereby proposes a DE-based Fuzzy Clustering Using Self-adaptive Trade-off Between Exploitation and Exploration (DEFSA). According to this research, self-adaptive trade-off between exploitation and exploration is remarkable such that none of arbitrarily setting parameters is used to control the mechanisms of such exploitation and exploration. The experiments are taken on six benchmarks real-world and two artificial data sets. The comparison tests are performed on the proposed method, DEFSA against a traditional DE-based fuzzy clustering, DE-based clustering as well as some other types of effective clustering algorithms such as SOM and AL.

MATERIALS AND METHODS

Combining Differential Evolution and Fuzzy C-Means (DEFCM): DE is a Relatively Heuristic algorithm which is designed to optimize problems over continuous domains. In DE, each decision variable is represented in the vector by a real number. As in any other Evolutionary algorithm, the initial population of DE is randomly generated and then evaluated. After that the mutation process takes place. During the mutation stage, three parents are chosen and they generate a single offspring which competes with a parent to determine who passes to the following generation. DE generates a single offspring by adding the weighted difference vector between two parents to a third parent. During crossover, each offspring and parent vectors participate for creation of trail vectors depending on Crossover Rate (CR). The CR has been provided by the user in the range [0, 1]. If the trail vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with respect to which it was compared.

Vector representation and population initialization: A population in DE process consists of several individual partitions P_i of K cluster centers, i = 1, ..., M, M is a population size and $P_i = [c_{i1}...c_{ik}]$ where c_{ij} refers to cluster center j in partition P_i, j = 1, ... K. cij = $[f^1_{ij}...f^d_{ij}...f^D_{ij}]^T$ where f^d_{ij} refers to feature d of cluster j in partition P_i and D is the number of features. Thus, one would determine as:

$$P_i \in \Re^{D\times K}$$

Where:

$$P_i = \left[f_{ij}^d\right]_{D\times k}$$

Fitness computation and finding the optimal partition: The fitness of an individual is computed based on fuzzy c-means objective function (Eq. 1):

$$\text{FCM\_objective function} = \sum_{n=1}^{N}\sum_{k=1}^{K}\mu_{nk}^{m}\left\|\mathbf{x}_n - \mathbf{c}_{ij}\right\|^2 \qquad (1)$$

μ_{nk} represents membership of sample x_n, n = 1, ..., N in cluster j. m = relates to a degree of fuzziness. For crisp data, μ_{nk} is one if x_n is in cluster j and is zero if not. Then, the partition with the optimal fitness, P^{opt} is chosen.

Mutation: Three particular partitions, $P_i \in \Re^{D\times K}$ are randomly picked up, named P_{r0}, P_{r1} and P_{r2} where r1, r2 and r3∈{1, ..., M}. Such three partitions, together with scaling factor, F∈[0, 1] are used to compute a mutant partition, P_i^{mut} as seen in Eq. 2:

$$P_i^{mut} = P_{r0} + F(P_{r1} - P_{r2}) \qquad (2)$$

Crossover: In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. P_i^{cross} = $[fcross^d_{ij}]_{D\times K}$ refers to the partition processed by crossover:

$$P_i^{mut} = \left[fmut_{ij}^d \right]_{D\times K}$$

And:

$$P^{opt} = [fopt_{ij}^d]_{D\times K}$$

where, $f\_cross_{ij}^d$, $F\_mut_{ij}^d$, respectively represent feature d of cluster j in partition i, processed by crossover, mutation and $f\_opt_j^d$ is feature d of cluster j in the optimal partition found in Eq. 3:

$$f\_cross_{ij}^d = \begin{cases} fmut_{ij}^d & \text{if } rand_d(0, 1) \le CR \text{ or } d = rand(j) \\ fopt_j^d & \text{if } rand_d(0, 1) > CR \text{ or } d \ne rand(j) \end{cases} \tag{3}$$

In Eq. 3, $rand_d(0, 1)$ is the dth evaluation of a uniform random number generator with outcome $\in[0, 1]$. CR is the crossover constant $\in[0, 1]$ which has to be determined by the user. rand(j) is a randomly chosen index $\in 1, 2, ..., D$ which ensures that $fcross_{ij}^d$ gets at least one parameter from $fmut_{ij}^d$.

Selection: To decide which individual partition should become a member of the next generation P_i^{cross} is compared to the optimal partition P^{opt} using greedy criterion. If P_i^{cross} yields better fitness value than P^{opt} then P_i^{cross} is set to P_i in the next generation, otherwise, the old value of is P_i retained.

Termination criterion: The processes of mutation, crossover and selection are executed for a fixed number of iterations. The best partition, P^{opt} seen up to the last generation of the population provides the solution to the clustering problem.

FCM clustering refinement: After DE clustering process, FCM is performed for refining the clusters using P^{opt} yielded in termination criterion as the initial partition.

A differential evolution-based fuzzy clustering using self-adaptive trade-off between exploitation and exploration (DEFSA): Drawbacks of most Optimization and Clustering algorithms including DEFCM concerns the low performance of driving exploitation or exploration operations during the search, this easily leads to local optimal traps or divergence of the search. DEFSA is introduced here to relieve such problems as a consequence, the global optimal solution can be accomplished. In the proposed method, efficiently dynamic exploitation and exploration mechanism is controlled by none of arbitrarily defined parameters. The main idea behind such proposed method is to exploit the search around a region in the gap between the nearby cluster centers in two current best partitions of cluster. Such a region tends to contain the better partition solutions in the coming iteration, nevertheless, the regions outside the gap are also explored for possible better solutions as well. An overview process of DEFSA is described in Fig. 1. Based on a randomly selected initial partition of clusters, DEFCM is executed. A couple of optimal partitions P_1^{opt} and P_2^{opt} which represent the best and secondary one is picked up instead of a single one. The feature values of both partitions are linearly normalized. Then, inside and outside gaps between the nearby cluster centers in those P_1^{opt} and P_2^{opt} are indicated. As aforementioned, the inside-gap refers to the exploitation area where the outside-gap one refers to the exploration area. The interval of such gaps is illustrated as an example in Fig. 2 and would be explained in detail later on. P_1^{opt} and P_2^{opt} and are randomly initialized within each gap regions and are independently fed as seeds to DEFCM. Afterwards P_1^{opt} and P_2^{opt} are achieved from those two regions by the DEFCM. Each of those partitions represents the optimal clusters found in the area of exploitation and exploration. Once again, inside and outside gaps between the nearby cluster centers in the new and are defined. The execution loop is repeated it would be terminated when the pre-defined number of iterations is reached or one of the optimal partitions can pass the setting performance criteria.

Figure 2a shows partitions P_1^{opt} and P_2^{opt} each of which successively composes of two clusters (K = 2) c_{11}^{opt}, c_{12}^{opt} and c_{21}^{opt}, c_{22}^{opt} with three features (D = 3). Based on Euclidean distance, 2 couples of the nearby clusters between the two partitions P_1^{new} and P_2^{new} are defined the first one is $c_{11}^{opt} = [0.1\ 0.8\ 0.4]^T$, $c_{22}^{opt} = [0.2\ 0.2\ 0.4]^T$ and the second one is $c_{12}^{opt} = [0.3\ 0.5\ 0.9]^T$ and $c_{21}^{opt} = [0.7\ 0.6\ 0.1]^T$. The inside-gap between the features in first couple of clusters are in the range [0.1, 0.2], [0.2, 0.8] and [0.4, 0.4], respectively. Likewise, the gap between the other couple are in [0.3, 0.7], [0.5, 0.6] and [0.1, 0.9]. These inside-gap regions are used for randomly creating the clusters in a new partition, P_1^{new} vice versa, the outside-gap ones based on the area outside those ranges are used for developing the clusters in P_2^{new}. Such regions are pointed in Fig. 2b. P_1^{new} composed of randomly selected cluster $c_{11}^{new} = [fnew_{11}^1\ fnew_{11}^2\ fnew_{11}^3]$ and $c_{12}^{new} = [fnew_{12}^1\ fnew_{12}^2\ fnew_{12}^3]$ where fnew $^d_{1j}$refers to feature d in the cluster c_{1j}^{new} belonging to P_1^{new}, j = 1, 2; the individual $fnew_{1j}^d$ are randomly selected based on these following ranges: $fnew_{11}^1\in[0.1, 0.2]$, $fnew_{11}^2\in[0.2, 0.8]$ and $fnew_{11}^3 = 0.4$ by the same way $fnew_{12}^1\in[0.3, 0.7]$, $fnew_{12}^2\in[0.5, 0.6]$ and $fnew_{13}^3\in[0.1, 0.9]$ It is noticed that those ranges are generated from the gap between c_{11} and

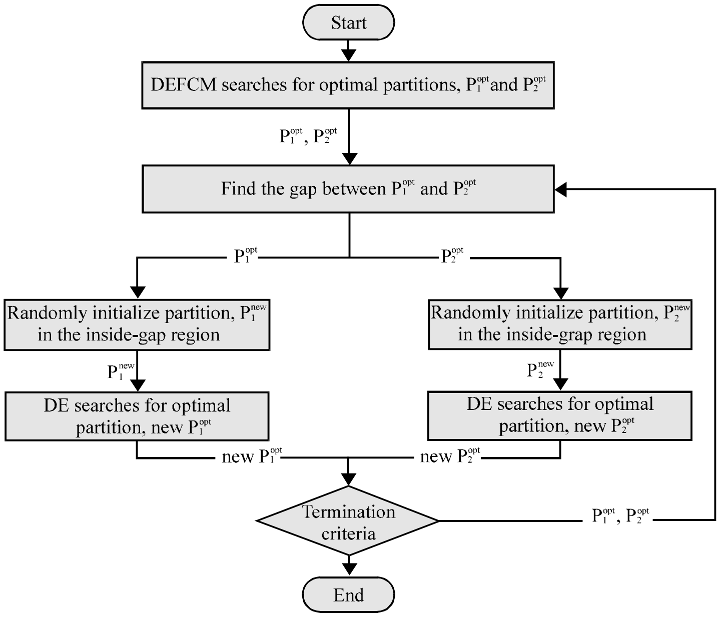
Start
DEFCM searches for optimal partitions, P_1^{opt} and P_2^{opt}
P_1^{opt}, P_2^{opt}
Find the gap between P_1^{opt} and P_2^{opt}
P_1^{opt}
P_2^{opt}
Randomly initialize partition, P_1^{new} in the inside-gap region
Randomly initialize partition, P_2^{new} in the inside-grap region
P_1^{new}
P_2^{new}
DE searches for optimal partition, new P_1^{opt}
DE searches for optimal partition, new P_2^{opt}
new P_1^{opt}
new P_2^{opt}
Termination criteria
P_1^{opt}, P_2^{opt}
End

Fig. 1: An overview process of the proposed method (DEFSA)

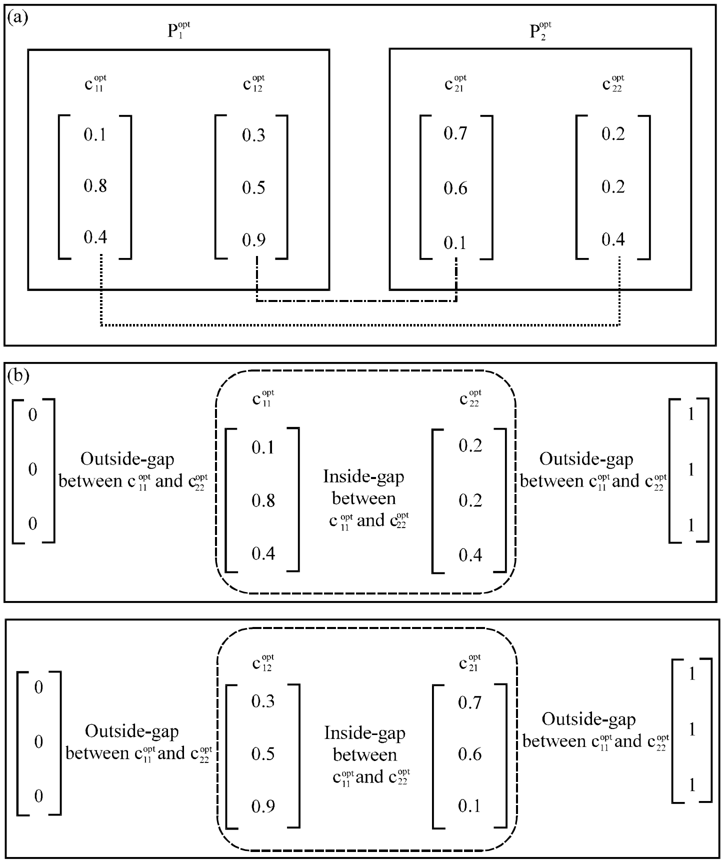
(a)
P_1^{opt}
P_2^{opt}
c_{11}^{opt}
c_{12}^{opt}
c_{21}^{opt}
c_{22}^{opt}
0.1
0.8
0.4
0.3
0.5
0.9
0.7
0.6
0.1
0.2
0.2
0.4
(b)
0
0
0
Outside-gap between c_{11}^{opt} and c_{22}^{opt}
c_{11}^{opt}
0.1
0.8
0.4
Inside-gap between c_{11}^{opt} and c_{22}^{opt}
c_{22}^{opt}
0.2
0.2
0.4
Outside-gap between c_{11}^{opt} and c_{22}^{opt}
1
1
1
0
0
0
Outside-gap between $c_{11}^{opt}$and c_{22}^{opt}
c_{12}^{opt}
0.3
0.5
0.9
Inside-gap between $c_{11}^{opt}$and c_{22}^{opt}
c_{21}^{opt}
0.7
0.6
0.1
Outside-gap between $c_{11}^{opt}$and c_{22}^{opt}
1
1
1

Fig. 2: a) An example of two partitions P_1^{opt} and P_2^{opt} composed of two clusters with three features for each and b) the region inside and outside the gaps between clusters used for exploitation and exploration of the optimal partition solutions

c_{21} as well as c_{12} and c_{22}. Subsequently, such P_1^{new} would be employed as the initial seed partition for further DEFCM optimization process. After a certain number of DEFCM iterations, the optimal partition of clusters, the new P_1^{opt} is discovered as a result; and it is supposed to be an optimal solution found in exploitation area. However, the region outside the gap is unexplored that leads to the imbalance between exploitation and exploration. By this reason, the search space outside the gap between the two old optimal partitions P_1^{opt} and P_2^{opt} should be also involved. Under the outside-gap regions, the partition P_2^{new} is developed. It composes of randomly selected clusters c_{21}^{new} = [$fnew_{21}^1$ $fnew_{21}^2$ $fnew_{21}^3$], c_{22}^{new} = [$fnew_{22}^1$ $fnew_{22}^2$ $fnew_{22}^3$] where $fnew_{2j}^d$ refers to feature d in the cluster c_{2j}^{new} belonging to P_2^{new} This outside-gap is as well illustrated along with the inside-gap in Fig. 2b. In contrast to the exploitation search which is performed under the inside-gap area, each of the features $fnew_{2j}^d$ in cluster c_{21}^{new} and c_{22}^{new} is randomly selected based on the following regions, $fnew_{21}^1 \notin [0.1, 0.2]$, $fnew_{21}^2 \notin [0.2, 0.8]$ and $fnew_{21}^3 \neq 0.4$ by the same way $fnew_{22}^1 \notin [0.3, 0.7]$, $fnew_{22}^2 \notin [0.5, 0.6]$ and $fnew_{22}^3 \notin [0.1, 0.9]$. Then, P_2^{new} would be submitted as the initial seed partition to DEFCM optimization process. The new optimal partition of clusters P_2^{opt} is obtained as a consequence and it represents the best partition from exploration side. If one of the new P_1^{opt} or P_2^{opt} can pass the setting performance criteria or the defined maximum iteration is reached, then the iteration is terminated, else those new P_1^{opt} and P_2^{opt} would be fed into the execution loop in the next iteration as indicated in Fig. 1. In cases of the number of clusters, $K \geq 2$ the similar process can still be applied. Additionally, the search procedure points out that in an execution loop, efficient exploitation and exploration mechanisms are dynamically pursued without any control none of arbitrarily setting parameters is needed to drive the mechanisms of the search.

Although, the searches are executed in both inside and outside gap based on two optimal partitions, P_1^{opt} and P_2^{opt} the worst-case complexity (big-O notation) relating to DEFSA is determined as $O(N \times K \times D \times M)$ where N, K, D and M refer to the number of sample cases, clusters, features and DE population size consecutively.

RESULTS

The tested data sets consist of two artificial data sets: Artset1 and Artset2 and six well-known real-world data sets, available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/named parkinson, hepatitis, dermatology, breast tissue, iris and wine. The characteristics of the real-world data sets are described in Table 1.

Table 1: Characteristics of data sets considered

Data sets	Number of features	Number of sample cases	Clusters
Parkinson	22	195 (48, 147)	2
Hepatitis	18	148 (2, 81, 61, 4)	4
Dermatology	34	366 (112, 61, 72, 49, 52, 20)	6
Breast tissue	9	106 (21, 15, 18, 16, 14, 22)	6
Iris	4	150 (50, 50, 50)	3
Wine	13	178 (59, 71, 48)	3

The artificial data set, Artset1 is a two-feature problem with three unique classes. A total of 900 patterns are drawn from three independent bivariate normal distributions where classes are distributed according to:

$$N_2\left(\mu = \begin{pmatrix} \mu_{i1} \\ \mu_{i2} \end{pmatrix}, \sum \begin{bmatrix} 0.080 & 0.076 \\ 0.076 & 0.074 \end{bmatrix}\right),$$

$$i = 1, 2, 3, \mu_{11} = 0.163, \mu_{12} = 0.147,$$

$$\mu_{21} = 0.535, \mu_{22} = 0.477, \mu 31 = 0.838, \mu_{32} = 0.799$$

Where:

$\begin{pmatrix} \mu_{i1} \\ \mu_{i2} \end{pmatrix}$ = Mean vector of class i

Σ = A covariance matrix

The data of Artset1 is illustrated in Fig. 3a. Figure 3b shows Artset2, a three-feature artificial data set with three classes and 300 patterns where the sample cases in each class is distributed in such a following manner:

$$\text{Class1} \sim \text{Uniform} \begin{vmatrix} 0.512 & 0.798 \\ 0.143 & 0.547 \\ 0.448 & 0.644 \end{vmatrix}$$

$$\text{Class2} \sim \text{Uniform} \begin{vmatrix} 0 & 0.275 \\ 0 & 0.490 \\ 0 & 0.247 \end{vmatrix}$$

$$\text{Class3} \sim \text{Uniform} \begin{vmatrix} 0.708 & 1 \\ 0.461 & 1 \\ 0.658 & 1 \end{vmatrix}$$

The proposed method, DEFSA and all comparative clustering methods: DEFCM, DE alone and two Efficient Clustering Methods, SOM and AL are evaluated and are implemented using MATLAB 7.10 (R2010a) on a CPU 2.4 GHZ Core2™Quad with 4 GB RAM. DEFSA as well as DEFCM employ 30 population, running 10 maximum iterations.

The results are refined by 300 FCM runs. DE also employs 30 populations on 10 maximum iteration runs. The 300 maximum iterations are consumed by FCM, SOM and AL.

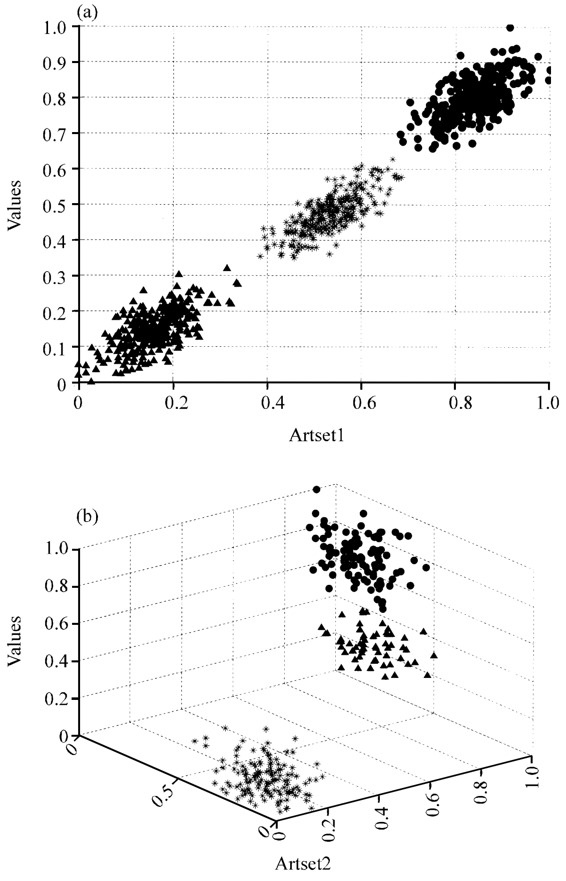

Fig. 3: The artificial data sets; a) Artset1 and b) Artset2

Such numbers of runs are assigned for fair comparison tests. Ten independent cross-validation runs are performed to generate average results for all the methods. The quality of the respective clustering approaches are evaluated and compared based on the following criteria measure.

The objective function values of FCM: This is the sum over all the distance from a sample case to all the centers as defined in Eq. 1. Clearly, the smaller the sum is the higher the quality of clustering would be.

The F-measure: This is related with the precision and the recall from the information retrieval (Dalli, 2003; Handl *et al*., 2003). The precision and the recall are defined as:

$$p(i,j) = \frac{n_{ij}}{n_i},\ r(i,j) = \frac{n_{ij}}{n_j} \quad (4)$$

Where each class i (given by the class labels of the used data set) is regarded as the set of n_i items desired for a query and each cluster j (generated by the algorithm) is regarded as the set of n_j items retrieved for a query. n_{ij} is the number of sample cases of the class i within cluster j. For a class i and a cluster j, the F-measure is defined as:

$$F(i,j) = \frac{(b^2+1).p(i,j).r(i,j)}{b^2.p(i,j).r(i,j)} \quad (5)$$

Where researchers choose b = 1 to obtain equal weighting for p(i, j) and r(i, j). The overall F-Measure for the data set of size n is given by:

$$F\text{-}\sum\nolimits_i \frac{n_i}{n_{\max_i}}\{F(i,j)\} \quad (6)$$

The bigger the F-measure is, the better the clustering algorithm is.

Xie-Beni index (XB): The XB (Xie and Beni, 1991; Olson, 1995) is called the compactness and separation validity function as shown in Eq. 7. The compactness and separation measure are respectively, indicated in numerator and denominator of the equation and are defined in Eq. 8 and 9. Small values of XB are expected for compact and well-separated clusters:

$$XB(C,X) = \frac{\sigma(C,X)}{n \times sep(C)} \quad (7)$$

$$\sigma(C,X) = \sum_{k=1}^{K}\sum_{i \in c^k} D^2(c^k, x_i) \quad (8)$$

$$sep(C) = \min_{i \neq K} \left\| x_i \text{-} c^k \right\|^2 \quad (9)$$

Where:

n	=	The number of sample cases
x_i	=	Sample case i
K	=	Total number of clusters
$D^2(c^k, x_i)$	=	A Euclidian distance between c^k and x_i
c^k	=	The center of cluster k

DISCUSSION

The derived boxplots in Fig. 4 signifies the competent F-measure degrees of the proposed method, DEFSA for all data sets. However, the proficiently low standard deviation of the F-measure degree of DEFSA is noticeable for both types of data. This is one of remarkable outcomes of DEFSA. Although, Table 2 shows a little bit inefficient runtime consumption, produced by DEFSA, Table 3 indicates superior results of DEFSA over

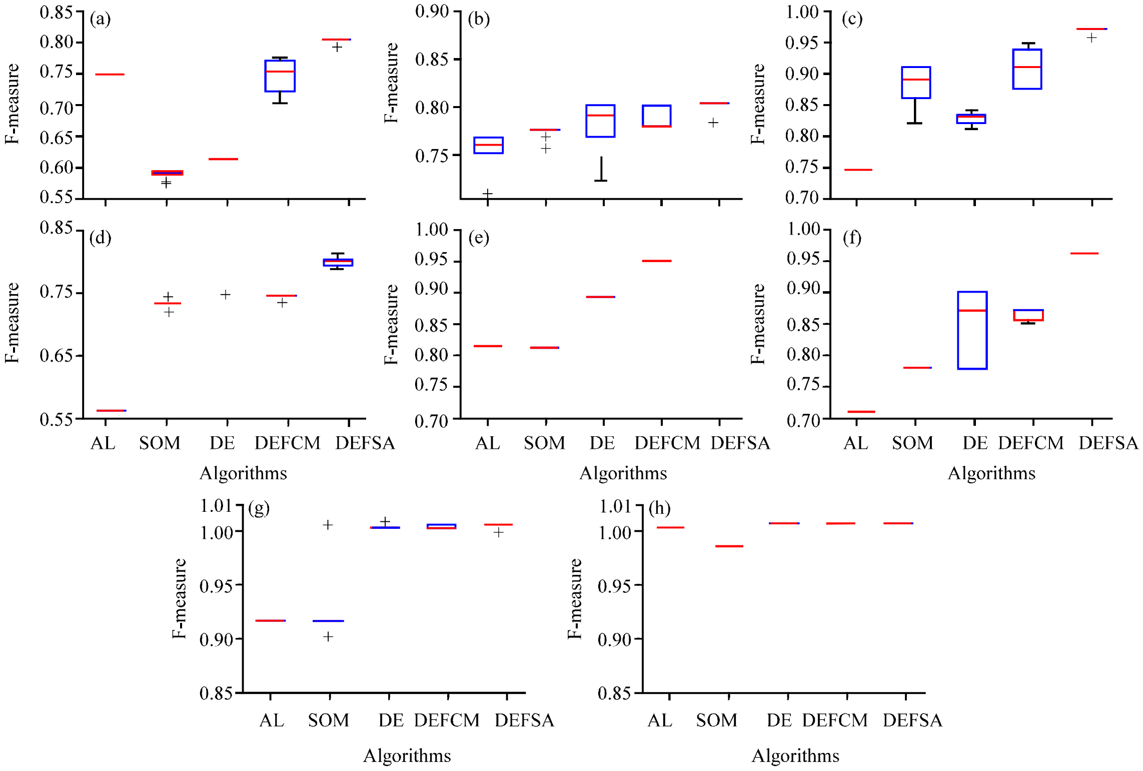

Fig. 4: Ranges of F-measure degrees, resulted from running the five clustering algorithms on the eight data sets; a) Parkinson; b) Hepatitis; c) Dermatology; d) Breast tissue; e) Iris; f) Wine; g) Artset1 and h) Artset2

Table 2: Runtimes in seconds, consumed by the six algorithms on the eight data sets

Source/Algorithms	DEFSA	DEFCM	DE	SOM	AL
Parkinson	0.2069 (0.0013)	0.2267 (0.0789)	**0.0102 (0.0147)**	0.9330 (0.0935)	0.1110 (0.2572)
Hepatitis	**0.1923 (0.0019)**	1.0340 (0.1052)	0.8929 (0.0472)	1.4637 (0.0259)	4.0115 (0.2039)
Dermatology	0.8149 (0.0442)	0.2653 (0.0148)	**0.0319 (0.0033)**	1.4241 (0.0418)	0.0620 (0.0161)
Breast tissue	0.3845 (0.0451)	1.3630 (0.0762)	**0.0186 (0.0085)**	1.8897 (0.0843)	0.0202 (0.0044)
Iris	0.1740 (0.0020)	0.3351 (0.0328)	0.0462 (0.0022)	**0.0062 (0.0012)**	0.8886 (0.0670)
Wine	0.8110 (0.2436)	0.9044 (0.4690)	**0.2567 (0.0553)**	2.1069 (0.7012)	3.7740 (0.9148)
Artset1	1.2343 (0.3358)	**0.0453 (0.0187)**	0.0590 (0.0386)	0.9232 (0.0328)	0.1333 (0.0135)
Artset2	1.3808 (0.2619)	0.0457 (0.0016)	**0.0148 (0.0041)**	0.9050 (0.0705)	0.1302 (0.0097)

Bold face indicates the best runtime

Table 3: F-Measure values, FCM objective degree and XB values resulted from running the six algorithms on the eight data sets

Source/Algorithms	DEFSA	DEFCM	DE	SOM	AL
Parkinson					
F-measure	**0.8060 (0.0025)**	0.7477 (0.0290)	0.6143 (0.0000)	0.59210 (0.0020)	0.7522 (0.0000)
FCM objective values	**15.2205 (3.0333)**	39.7225 (9.9309)	25.0486 (0.0022)	26.7978 (0.1202)	39.7165 (9.9309)
XB	1.9237 (0.2483)	0.7071 (0.3310)	**0.4208 (0.0050)**	0.84600 (0.0770)	0.6081 (0.4656)
Hepatitis					
F-measure	**0.7994 (0.0084)**	0.7883 (0.0107)	0.7780 (0.0308)	0.77410 (0.0032)	0.7507 (0.0224)
FCM objective values	**19.6489 (3.0139)**	25.5466 (0.0000)	28.5466 (0.0001)	28.5466 (0.0001)	25.5466 (0.0000)
XB	**1.6353 (0.1786)**	3.0354 (0.0871)	2.0513 (0.0003)	3.05160 (0.0008)	3.0514 (0.0001)
Dermatology					
F-measure	**0.9618 (0.0000)**	0.9109 (0.0276)	0.8294 (0.0089)	0.88350 (0.0307)	0.7483 (0.0000)
FCM objective values	**19.2166 (0.6867)**	21.1676 (1.6497)	19.9415 (0.0000)	22.1262 (0.6770)	21.1676 (1.6497)
XB	**1.6905 (0.0568)**	5.9994 (0.9800)	2.0076 (0.0013)	5.0366 (2.95580)	6.3875 (3.1366)
Breast tissue					
F-measure	**0.8803 (0.0101)**	0.8060 (0.0045)	0.7979 (0.0041)	0.78850 (0.0041)	0.5654 (0.0000)
FCM objective values	**0.4789 (0.0453)**	0.5431 (0.1678)	1.8233 (0.0482)	1.38580 (0.0049)	1.6312 (0.0000)
XB	0.5062 (0.1909)	0.5700 (0.0715)	0.3970 (0.0145)	**0.35340 (0.1845)**	1.3439 (0.0000)

Table 3: Continue

Source/Algorithms	DEFSA	DEFCM	DE	SOM	AL
Iris					
F-measure	**0.9825 (0.0000)**	0.9495 (0.0000)	0.8923 (0.0000)	0.8111 (0.0000)	0.8153 (0.0000)
FCM objective values	**1.8371 (0.2429)**	1.8668 (0.3055)	2.8842 (0.4552)	3.6518 (0.0001)	1.5191 (0.0058)
XB	0.5723 (0.0107)	0.2149 (0.0863)	0.2436 (0.1509)	0.4566 (0.0001)	**0.1410 (0.0524)**
Wine					
F-measure	**0.9613 (0.0000)**	0.8659 (0.0073)	0.8464 (0.0579)	0.7805 (0.0000)	0.7099 (0.0000)
FCM objective values	**16.6178 (2.6296)**	33.9205 (0.0001)	50.5369 (5.7725)	221.7619 (0.0002)	82.7330 (0.3629)
XB	1.1823 (0.0132)	**0.6008 (0.0001)**	0.6659 (0.2823)	0.6258 (0.0000)	0.6921 (0.1461)
Artset1					
F-Measure	**0.9994 (0.0013)**	0.9982 (0.0015)	0.9976 (0.0013)	0.9305 (0.0368)	0.9145 (0.0000)
FCM objective values	**5.0368 (0.3895)**	6.9644 (0.2183)	6.4631 (0.9912)	6.5053 (0.3269)	7.5364 (3.2095)
XB	**1.1504 (0.0058)**	2.1260 (0.0370)	2.1387 (0.0154)	1.8656 (0.0726)	1.2818 (0.0271)
Artset2					
F-measure	**1.0000 (0.0001)**	0.9999 (0.0001)	0.9999 (0.0001)	0.9804 (0.0000)	0.9970 (0.0000)
FCM objective values	**4.5263 (0.0980)**	4.6066 (0.8736)	6.3832 (0.0000)	5.9478 (0.0046)	8.9743 (2.8256)
XB	**0.1398 (0.0015)**	0.3311 (0.0544)	0.2356 (0.0220)	0.2435 (0.1150)	0.5652 (0.0512)

The mean and standard deviations (in parenthesizes) for 10 independent cross validation runs are reported on the top of the bars

the Comparative algorithms in terms of means and standard deviations (in parenthesizes) with respect to natural logarithmic values of FCM objective for all cases of data it also points the better XB degree for most cases. This confirms the efficiency of the DEFSA regarding both minimum dissimilarity within a cluster and maximum separateness between different clusters. However, the prominent advantage of DEFSA is that none arbitrarily control parameters are required for running exploration and exploitation. In consequence of these an efficient dynamic search has been done.

CONCLUSION

This study presents a differential evolution based fuzzy clustering using self-adaptive trade-off between exploitation and exploration (DEFSA). The main objective is to relieve the problems regarding divergence and local optimal traps, aiming to end up with the global optimal solution. The experiments are taken on six benchmarks real-world and two artificial data sets. The comparison tests are performed on the proposed method, DEFSA against the former DEFCM clustering, DE, clustering alone as well as two effective clustering algorithms, SOM and AL. Among all comparative clustering methods, the proposed DEFSA reports the highest encouraging results in terms of F-measure and FCM objective degrees. The important merit of DEFSA is that it can generate an efficient dynamic exploitation and exploration mechanism without any arbitrarily defined trading-off parameter. The idea behind this is to heuristically consider the gap between two current best cluster partitions as the exploitation area and the rest outside the gap as the exploration one. However, the runtime is not much attractive. Future research may focus on some other methodologies of discovering the proper search space for exploration and exploitation using only an individual partition to achieve better runtime results.

REFERENCES

Alpaydm, E., 2004. Introduction to Machine Learning. The MIT Press, Cambridge, MA., USA., ISBN-13: 9780262012119, pp: 1-3.

Aydin, D., 2007. Comparison of regression models based on nonparametric estimation techniques: Prediction of GDP in Turkey. Int. J. Math. Models Methods Applied Sci., 1: 70-75.

Bezdek, J.C., 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, USA., ISBN-13: 978-1475704525, Pages: 256.

Civicioglu, P. and E. Besdok, 2013. A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms. Artif. Intell. Rev., 39: 315-346.

Dalli, A., 2003. Adaptation of the F-measure to cluster-based Lexicon quality evaluation. Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, April 12-17, 2003, Agro Hotel, Budapest, Hungary, pp: 51-56.

Dogan, B. and M. Korurek, 2012. A new ECG beat clustering method based on kernelized fuzzy c-means and hybrid ant colony optimization for continuous domains. Applied Soft Comput., 12: 3442-3451.

Dunn, J.C., 1973. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybernet., 3: 32-57.

Han, M.F., C.T. Lin and J.Y. Chang, 2013. Efficient differential evolution algorithm-based optimisation of fuzzy prediction model for time series forecasting. Int. J. Intell. Inform. Database Syst., 7: 225-241.

Handl, J., J. Knowles and M. Dorigo, 2003. On the performance of ant-based clustering. Proceedings of the 3rd International Conference on Hybrid Intelligent Systems, December 14-17, 2003, Melbourne, Australia, pp: 204-213.

Hastie, T., R. Tibshirani and J. Friedman, 2009. Hierarchical Clustering. In: The Elements of Statistical Learning: Data Mining, Inference and Prediction, Hastie, T., R. Tibshirani and J. Friedman (Eds.). 2nd Edn., Springer, New York, USA., ISBN-13: 9780387848587, pp: 520-528.

Herrero, A., E. Corchado and J. Alfredo, 2011. Unsupervised neural models for country and political risk analysis. Expert Syst. Appl., 38: 13641-13661.

Izakian, H. and A. Abraham, 2011. Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. Expert Syst. Appl., 38: 1835-1838.

Kao, Y. and C.C. Chen, 2013. A differential evolution fuzzy clustering approach to machine cell formation. Int. J. Adv. Manuf. Technol., 65: 1247-1259.

Kohonen, T., 1995. Self-Organizing Maps. 1st Edn., Springer, New York, USA.

Li, X., C. Hu and X. Yan, 2013. Chaotic differential evolution algorithm based on competitive coevolution and its application to dynamic optimization of chemical processes. Intell. Autom. Soft Comput., 19: 85-98.

Memmedli, M. and A. Nizamitdinov, 2013. An application of various nonparametric techniques by nonparametric regression splines. Int. J. Math. Models Methods Applied Sci., 6: 106-113.

Montgomery, J., M. Randall and A. Lewis, 2011. Differential evolution for RFID antenna design: A comparison with ant colony optimisation. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, July 12-16, 2011, Dublin, Ireland, pp: 673-680.

Olson, C.F., 1995. Parallel algorithms for hierarchical clustering. Parallel Comput., 21: 1313-1325.

Pai, D.R., K.D. Lawrence, R.K. Klimberg and S.M. Lawrence, 2012. Experimental comparison of parametric, non-parametric and hybrid multigroup classification. Expert Syst. Appl., 39: 8593-8603.

Ravi, V., N. Aggarwal and N. Chauhan, 2010. Differential evolution based fuzzy clustering. Proceedings of the 1st International Conference on Swarm, Evolutionary and Memetic Computing, December 16-18, 2010, Chennai, India, pp: 38-45.

Rojas, R., 1996. Neural Networks: A Systematic Introduction. Springer-Verlag, Berlin, Germany, ISBN-13: 9783540605058, Pages: 502.

Sayah, S., A. Hamouda and K. Zehar, 2013. Economic dispatch using improved differential evolution approach: A case study of the Algerian electrical network. Arabian J. Sci. Eng., 38: 715-722.

Slowik, A., 2011. Fuzzy control of trade-off between exploration and exploitation properties of evolutionary algorithms. Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems, May 23-25, 2011, Wroclaw, Poland, pp: 59-66.

Storn, R. and K. Price, 1997. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim., 11: 341-359.

Tan, P., M. Steinbach and V. Kumar, 2004. Introduction to Data Mining. Addison-Wesley, USA., pp: 1-35.

Tin, H.W., S.W. Leu and S.H. Chang, 2012. An PSO-based approach to speed up the fractal encoding. Int. J. Math. Models Methods Applied Sci., 6: 499-506.

Vucetic, D. and S.P. Simonovic, 2013. Evaluation and application of fuzzy differential evolution approach for benchmark optimization and reservoir operation problems. J. Hydroinform., 15: 1456-1473.

Wang, C.X., C.H. Li, H. Dong and F. Zhang, 2013. An efficient differential evolution algorithm for function optimization. Inform. Technol. J., 12: 444-448.

Wang, X. and S. Zhao, 2013. Differential evolution algorithm with self-adaptive population resizing mechanism. Math. Problems Eng. 10.1155/2013/419372.

Webb, A.R., 2002. Statistical Pattern Recognition. 2nd Edn., John Wiley and Sons Ltd., UK., ISBN-13: 978-0470845134, pp: 1-31.

Xie, X.L. and G. Beni, 1991. A validity Measure for fuzzy clustering. IEEE Trans. Pattern Anal. Mach. Intell., 13: 841-847.