Applied Soft Computing 11 (2011) 2765–2776

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A new multi-objective technique for differential fuzzy clustering

Indrajit Saha [a,*], Ujjwal Maulik [b], Dariusz Plewczynski [a]

[a] Interdisciplinary Centre for Mathematical and Computational Modeling, University of Warsaw, 02-106 Warsaw, Poland
[b] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, West Bengal, India

A R T I C L E I N F O

Article history:
Received 13 August 2009
Received in revised form 14 June 2010
Accepted 17 November 2010
Available online 24 November 2010

Keywords:
Fuzzy clustering
Modified differential evolution
Multiobjective optimization
Pareto-optimal
Cluster validity measures

A B S T R A C T

This article presents a new multiobjective differential evolution based fuzzy clustering technique. Recent research has shown that clustering techniques that optimize a single objective may not provide satisfactory result because no single validity measure works well on different kinds of data sets. The fact motivated us to present a new multiobjective Differential Evolution based fuzzy clustering technique that encodes the cluster centres in its vectors and optimizes multiple validity measures simultaneously. In the final generation, it produces a set of non-dominated solutions, from which the user can relatively judge and pick up the most promising one according to the problem requirements. Superiority of the proposed method over its single objective versions, multiobjective version of classical differential evolution and genetic algorithm, well-known fuzzy C-means and average linkage clustering algorithms has been demonstrated quantitatively and visually for several synthetic and real life data sets. Statistical significance test has been conducted to establish the statistical superiority of the proposed multiobjective clustering approach. Finally, the proposed algorithm has been applied for segmentation of a remote sensing image to show its effectiveness in pixel classification.

1. Introduction

Clustering [1–4] is a useful unsupervised data mining technique which partitions the input space into K regions depending on some similarity/dissimilarity metric where the value of K may or may not be known *a priori*. The main objective of any clustering technique is to produce a $K \times n$ partition matrix $U(X)$ of the given data set X, consisting of n patterns, $X=\{x_1, x_2, \ldots, x_n\}$. The partition matrix may be represented as $U=[u_{k,j}]$, $k=1, \ldots, K$ and $j=1, \ldots, n$, where $u_{k,j}$ is the membership of pattern x_j to the kth cluster. For fuzzy clustering of the data, $0<u_{k,j}<1$, i.e., $u_{k,j}$ denotes the degree of belongingness of pattern x_j to the k th cluster. Evolving an appropriate $U(X)$ is a complex optimization problem, requiring the application of advance optimization techniques.

Recently, the application of differential evolution (DE) [5–7] in the field of clustering has attracted the attention of researchers and it has been effectively used to develop single objective clustering techniques [8–10]. However, a single cluster validity measure is seldom equally applicable for different kinds of data sets with different characteristics. Hence, it is necessary to simultaneously optimize several validity measures that can capture the different data characteristics. In order to achieve this, in this paper, the problem of fuzzy partitioning is posed as one of the multiobjective optimizations (MOOs) [11–13], where a search is performed over a number of, often conflicting, objective functions. The final solution set contains a number of Pareto-optimal solutions, none of which can be further improved on any one objective without degrading it in another. Such that, real-coded multiobjective modified differential evolution based fuzzy clustering (MOMoDEFC) is proposed in this regard in order to determine the appropriate cluster centres and the corresponding partition matrix. The Xie-Beni (XB) index [14] and the fuzzy C-means (FCM) [15–17] measure (J_m) are used as the objective functions. Note that any other and any number of objective functions could be used instead of the above mentioned two.

Clustering results are reported for a number of artificial and real-life data sets. The efficiency of the proposed algorithm has also been verified in image segmentation. Indian remote sensing (IRS) satellite images of parts of the city of Calcutta and Bombay have been used for demonstrating the effectiveness of the developed MOMoDEFC technique in automatically segmenting the image into a known number of regions. Comparison with the well known fuzzy C-means algorithm [15,16] (which directly optimizes the FCM criterion), the single objective version of the modified differential evolution based fuzzy clustering [9] method that minimizes the J_m and XB indices separately, named as MoDEFC(J_m) and MoDEFC(XB), respectively, multiobjective version of classical differential evolution based fuzzy clustering (MODEFC) [18] and multiobjective genetic algorithm (non-dominated sorting genetic algorithm-II [19,11]) based fuzzy clustering (MOGAFC) [20] as well as a widely

* Corresponding author. Tel.: +48535251595.
E-mail addresses: indra@icm.edu.pl, indra_raju@yahoo.co.in (I. Saha), umaulik@cse.jdvu.ac.in (U. Maulik), darman@icm.edu.pl (D. Plewczynski).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.11.007

used hierarchical clustering algorithm (average-linkage) [21] are also provided.

The article is organized as follows: the next section provides the basic motivation and contribution of the proposed technique. In Section 3, some well-known algorithms for data clustering is described. Section 4 briefly describes single objective modified differential evolution based fuzzy clustering technique. Section 5 discuss the principal of multiobjective optimization. Subsequently Section 6 presents the proposed multiobjective clustering in detail. Section 7 presents the experimental results conducted on several synthetic and real life data sets along with Wilcoxon's rank sum test. In Section 8, the application of the proposed technique to satellite image classification is demonstrated. Finally Section 9 concludes the article.

2. Motivation and contribution

To solve many real world problems, it is necessary to optimize more than one objective simultaneously. Clustering is an important real world problem and different clustering algorithms usually attempt to optimize some validity measure such as the compactness of the clusters, separation among the clusters or combination of both. However, as the relative importance of different clustering criteria is unknown, it is better to optimize compactness and separation separately rather than combining them in a single measure to be optimized. Motivated by this fact, this article proposes a real-coded multiobjective modified differential evolution based fuzzy clustering (MOMoDEFC) algorithm that simultaneously optimizes both the compactness and the separation of the clusters, rather than the combination of them. In this regard, a well modified version of differential evolution [9] has been used as the underlying framework.

Multi-objective optimization has been gaining popularity since the last few years. There are some instances in literature that applied multi-objective techniques for data clustering. One of the earliest approaches in this field is found in [22] where objective functions representing compactness and separation of the clusters were optimized in a crisp clustering context and with a deterministic method. In [23], a tabu search based multiobjective clustering technique has been proposed, where the partitioning criteria are chosen as the within cluster similarity and between cluster dissimilarity. This technique uses solution representation based on cluster centres as in [24]. However experiments are mainly based on artificial distance matrices. A series of works on multi-objective clustering has been proposed in [25–27] where the authors have adopted chromosome encoding of length equal to the number of data points. The two objectives that were optimized are overall deviation (compactness) and connectivity. The algorithm in [25] is capable of handling categorical data, whereas, the other two papers deal with numeric and continuous data sets. These methods have advantages that they can automatically evolve the number of clusters and also can be used to find nonconvex shaped clusters. It may be noted that the chromosome length in these works is equal to the number of points to be clustered. Hence as discussed in [28], when the length of the chromosomes becomes equal to the number of points, n, to be clustered, the convergence becomes slower for the large values of n. This is due to the reason that the chromosomes, and hence the search space, in such cases become large. Moreover, these algorithms need special initialization routines based on minimum spanning tree method and are intended for crisp clustering. In contrast, the method proposed in this article uses a centre based encoding strategy for fuzzy clustering of numeric and continuous data sets. As fuzzy clustering is better equipped to handle overlapping clusters [28], the proposed technique can handle both overlapping and non-overlapping clusters.

The most important contribution of this article that it proposes a fuzzy multi-objective algorithm for clustering using well developed modified differential evolution. None of the previous works have addressed the issue of multi-objective fuzzy clustering. Two fuzzy objective functions, i.e., fuzzy compactness and fuzzy separation have been simultaneously optimized resulting in a set of non-dominated solutions.

3. Clustering algorithms and validity measures

This section describes two well-known clustering methods and some of the well-define internal as well as external cluster validity measures.

3.1. Clustering algorithms

3.1.1. Fuzzy C-means

Fuzzy C-means (FCM) [15,29] is a widely used technique that uses the principles of fuzzy sets to evolve a partition matrix $U(X)$ while minimizing the measure

$$J_m = \sum_{j=1}^{n}\sum_{k=1}^{K} u_{k,j}^{m} D^2(z_k, x_j) \tag{1}$$

where n is the number of data objects, K represents number of clusters, u is the fuzzy membership matrix (partition matrix) and m denotes the fuzzy exponent. Here x_j is the jth data point and z_k is the centre of kth cluster, and $D(z_k, x_j)$ denotes the distance of point x_j from the centre of the kth cluster.

FCM algorithm starts with random initial K cluster centres, and then at every iteration it finds the fuzzy membership of each data points to every cluster using the following equation

$$u_{k,i} = \frac{(1/D(z_k, x_i))^{1/(m-1)}}{\sum_{j=1}^{k}\left(1/D(z_j, x_i)\right)^{1/(m-1)}}, \quad \text{for } 1 \le k \le K,\ 1 \le i \le n \tag{2}$$

for $1 \le k \le K$; $1 \le i \le n$, where $D(z_k, x_i)$ and $D(z_j, x_i)$ are the distances between x_i and z_k, and x_i and z_j, respectively. The value of m, the fuzzy exponent, is taken as 2. Note that while computing $u_{k,i}$ using Eq. (2), if $D(z_j, x_i)$ is equal to zero for some j, then $u_{k,i}$ is set to zero for all $k = 1, \ldots, K$, kj, while $u_{k,i}$ is set equal to one. Based on the membership values, the cluster centres are recomputed using the following equation

$$z_k = \frac{\sum_{i=1}^{n} u_{k,i}^{m} x_i}{\sum_{i=1}^{N} u_{k,i}^{m}} \quad 1 \le k \le K \tag{3}$$

The algorithm terminates when there is no further change in the cluster centres. Finally, each data point is assigned to the cluster to which it has maximum membership. The main disadvantages of the fuzzy K-means clustering algorithms are (1) it depends much on the initial choice of the centres and (2) it often gets trapped into some local optimum.

3.1.2. Average-linkage clustering

The average-linkage (AL) clustering algorithm, also known as the unweighted pair-group method using arithmetic averages (UPGMA) [21], is one of the most widely used hierarchical clustering algorithms. The average-linkage algorithm is obtained by defining the distance between two clusters to be the average distance between a point in one cluster and a point in the other cluster. Formally, if C_i is a cluster with n_i members and C_j is a cluster with n_j members, the distance between the clusters is

$$\mathcal{D}_{AL}(C_i, C_j) = \frac{1}{n_i n_j} \sum_{a \in C_i, b \in C_j} D(a, b) \tag{4}$$

3.2. Internal cluster validity indices

For evaluating the performance of the clustering algorithms as an internal cluster validity measure, the Minkowski score [30] and adjusted rand index [31] are used. Details of these two indices [32,33] are given below.

3.2.1. Minkowski score

The performances of the clustering algorithms are evaluated in terms of the *Minkowski score* (MS) [30]. A clustering solution for a set of n elements can be represented by an $n \times n$ matrix C, where $C_{i,j} = 1$ if point i and j are in the same cluster according to the solution, and $C_{i,j} = 0$ otherwise. The Minkowski score of a clustering result C with reference to T, the matrix corresponding to the true clustering, is defined as

$$\mathrm{MS}(T, C) = \frac{||T - C||}{||T||} \tag{5}$$

where

$$||T|| = \sqrt{\sum_i \sum_j T_{i,j}}$$

The Minkowski score is the normalized distance between the two matrices. Lower Minkowski score implies better clustering solution, and a perfect solution will have a score zero.

3.2.2. Adjusted rand index

Suppose T is the true clustering of a data set based on domain knowledge and C a clustering result given by some clustering algorithm. Let a, b, c and d, respectively denote the number of pairs belonging to the same cluster in both T and C, the number of pairs belonging to the same cluster in T but to different clusters in C, the number of pairs belonging to different clusters in T but to the same cluster in C and the number of pairs belonging to different clusters in both T and C. The adjusted rand index ARI(T, C) is then defined as follows:

$$\mathrm{ARI}(T, C) = \frac{2(ad - bc)}{(a + b)(b + d) + (a + c)(c + d)} \tag{6}$$

The value of ARI(T, C) lies between 0 and 1 and higher value indicates that C is more similar to T. Also, ARI(T, T) = 1.

3.3. External cluster validity indices

Xie-Beni [14] and $\Im$ index [34] are used for external cluster validity measure. These two indices are described in below.

3.3.1. Xie-Beni index

The XB index [14] is defined as a function of the ratio of the total variation σ to the minimum separation *sep* of the clusters. Here σ and *sep* can be written as

$$\sigma(U, Z; X) = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{k,i}^2 D^2(z_k, x_i) \tag{7}$$

and

$$sep(Z) = \min_{i \neq j} ||z_k - z_j||^2 \tag{8}$$

where $||\cdot||$ is the Euclidean norm, and $D(z_k, x_i)$, as mentioned earlier, is the distance between the pattern x_i and the cluster centre z_k. The XB index is then define as

$$\mathrm{XB}(U, Z; X) = \frac{\sigma(U, Z; X)}{n \times sep(Z)} \tag{9}$$

Note that when the partitioning is compact and good, value of σ should be low while *sep* should be high, thereby yielding lower values of the Xie-Beni (XB) index. The objective is therefore to minimize the XB index for achieving proper clustering.

3.3.2. $\Im$ *index*

A cluster validity index $\Im$, proposed in [34] is defined as follows:

$$\Im(K) = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_K\right)^p, \tag{10}$$

where K is the number of clusters. Here,

$$E_K = \sum_{k=1}^{K} \sum_{i=1}^{n} u_{k,i} ||z_k - x_i|| \tag{11}$$

and

$$D_K = \max_{i \neq j} ||z_k - z_j|| \tag{12}$$

The index $\Im$ is a composition of three factors, namely, $1/K$; E_1/E_K and D_K. The first factor will try to reduce index $\Im$ as K is increased. The second factor consists of the ratio of E_1, which is constant for a given data set, to E_K, which decreases with increase in K. To compute E_1 the value of K in Eq. (11) is taken as one i.e., all the points are considered to be in the same cluster. Note that index $\Im$ increases as E_K decreases. This, in turn, indicates that formation of more number of clusters, which are compact in nature, would be encouraged. Finally, the third factor, D_K (which measures the maximum separation between two clusters over all possible pairs of clusters), will increase with the value of K. However, note that this value is upper bounded by the maximum separation between two points in the data set. Thus, the three factors are found to compete with and balance each other critically. The power p is used to control the contrast between the different cluster configurations. In this article, we have taken $p = 2$.

4. Modified differential evolution based fuzzy clustering

In this section, the principle of modified differential evolution (MoDE) [9] and modified differential evolution based fuzzy clustering (MoDEFC) [9] has been discussed.

4.1. Modified differential evolution

In modified differential evolution (MoDE) an approach has been introduced during mutation to push the trial vector quickly towards the global optima. The crossover and selection are same as original DE. In the mutation process, for creating the ith mutant offspring vector other three unique vectors are used. Two of them are represented the global best (*GBest*, best vector evaluated till the current generation) and local best (*LBest*, best vector in current population), whereas third one is the jth vector of the current population. There are two different mutation processes. In each generation, (denoted by t) one of the process is selected depending on α (alpha), computed as $1/(1 + \exp(-(1/\text{generation})))$. Thus one process uses the mutation as in original DE, other generates a new mutant vector using Eq. (13). Note that as number of generations increases the value of α decreases in the range between [1, 0.5]. It results in a lower probability of using modi*Mutation*. Also when modi*Mutation* is used in the initial stage, *LBest* has contributed more for evolving the mutant vector than in the later stage. As the contribution of *LBest* for the mutant vector decreases with generation, contribution of *GBest* increases. Fig. 1 describes the different steps of MoDE algorithm.

$$G_i(t + 1) = G_{GBest}(t) + \alpha(G_{LBest}(t) - G_{(j)}(t)) \tag{13}$$

1. Initialization
2. Evaluation
3. Set GBest and LBest

Repeat

alpha=1/(1+exp(-(1/generation)))

if(rand(0,1)<alpha)

4. modiMutation

else

4. Mutation

end if

5. Crossover
6. Evaluation
7. Update LBest

if(LBest<GBest)

8. Set GBest with LBest

end if

9. Selection

Until (termination criteria are met)

Fig. 1. Algorithm of MoDE.

4.2. Modified differentia evolution bsed fuzzy clustering algorithm

In modified differentia evolution bsed fuzzy clustering (MoDEFC) each vector is a sequence of real numbers representing the K cluster centres [24]. For an d-dimensional space, the length of a vector is $l = d \times K$, where the first d positions represent the first cluster centre, the next d positions represent those of the second cluster centre, and so on. The K cluster centres encoded in each vector are initialized to K randomly chosen points from the data set. This process is repeated for each of the P vectors in the population, where P is the size of the population. In this article, either the FCM measure or the XB index is taken as the objective that needs to be optimized. Subsequently, the centres encoded in a vector are updated using the Eq. (2). The ith individual vector of the population at time-step (generation) t has l components, i.e.,

$$G_{i,l}(t) = [G_{i,1}(t), G_{i,2}(t), \ldots, G_{i,l}(t)] \tag{14}$$

For each target vector $G_{i,l}(t)$ that belongs to the current population, we use *alpha* as a criteria on favor of MoDE that randomly samples three other individuals, i.e., $G_{Lbest,l}(t)$, $G_{Gbest,l}(t)$ and $G_{m,l}(t)$, (as describe earlier) or otherwise $G_{k,l}(t)$, $G_{n,l}(t)$ and $G_{m,l}(t)$ from the same generation. Then calculates the (componentwise) difference, scales it by a scalar α or F (usually $\in[0,\ 1]$), and creates a mutant offspring $V_{i,l}(t+1)$.

$$V_{i,l}(t+1) = \begin{cases} G_{Gbest,l}(t) + \alpha(G_{Lbest,l}(t) - G_{m,l}(t)), & \text{if } rand(0,1) < \alpha \\ G_{k,l}(t) + F(G_{n,l}(t) - G_{m,l}(t)), & \text{otherwise} \end{cases} \tag{15}$$

In order to increase the diversity of the perturbed parameter vectors, crossover is introduced. To this end, the trial vector:

$$U_{i,l}(t+1) = [U_{i,1}(t+1), U_{i,2}(t+1), \ldots, U_{i,l}(t+1)] \tag{16}$$

is formed, where

$$U_{i,l}(t+1) = \begin{cases} V_{i,l}(t+1) & \text{if } rand_l(0,1) \le \text{CR or } l = rand(i) \\ G_{i,l}(t) & \text{if } rand_l(0,1) > \text{CR and } l rand(i) \end{cases} \tag{17}$$

In Eq. (17), $rand_l(0,\ 1)$ is the lth 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 $U_{i,l}(t+1)$. $rand(i)$ is a randomly chosen index $\in 1, 2, \ldots, l$ which ensures that α gets at least one parameter from $V_{i,l}(t+1)$. To decide whether or not it should become a member of generation $t+1$, the trial vector $U_{i,l}(t+1)$ is compared to the target vector $G_{i,l}(t)$ using the greedy criterion. If vector J_m yields a smaller cost function value than $G_{i,l}(t)$, then $U_{i,l}(t+1)$ is set to $G_{i,l}(t)$; otherwise, the old value J_m is retained. The processes of mutation, crossover and selection are executed for a fixed number of iterations. The best vector seen up to the last generation provides the solution to the clustering problem.

5. Multiobjective optimization

In many real world situations there may be several objectives that must be optimized simultaneously in order to solve a certain problem. The main difficulty in considering multiobjective optimization is that there is no accepted definition of optimum in this case, and therefore it is difficult to compare one solution with another. In general, these problems admit multiple solutions, each of which is considered acceptable and equivalent when the relative importance of the objectives is unknown. The best solution is often subjective and depends on the need of the designer or decision maker.

The multiobjective optimization can be formally stated as following [12]: find the vector $\bar{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ of decision variables which will satisfy the m inequality constraints:

$$g_i(\bar{x}) \ge 0, \quad i = 1, 2, \ldots, m, \tag{18}$$

the p equality constraints

$$h_i(\bar{x}) \ge 0, \quad i = 1, 2, \ldots, p, \tag{19}$$

and optimizes the vector function

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_k(\bar{x})]^T \tag{20}$$

The constraints given in Eqs. (18) and (19) define the feasible region F which contains all the admissible solutions. Any solution outside this region is inadmissible since it violates one or more constraints. The vector $\bar{x}^*$ denotes an optimal solution in F. In the context of multiobjective optimization, the difficulty lies in the definition of optimality, since it is only rare that we will find a situation where a single vector $\bar{x}^*$ represents the optimum solution to all the objective functions.

The concept of *Pareto optimality* comes handy in the domain of multiobjective optimization. A formal definition of Pareto optimality from the viewpoint of minimization problem may be given as follows: a decision vector $\bar{x}^*$ is called Pareto optimal if and only if there is no J_m that dominates $\bar{x}^*$, i.e., there is no $\bar{x}$ such that

$$\forall i \in \{1, 2, \ldots, k\}, \quad f_i(\bar{x}) \le f_i(\bar{x}^*) \tag{21}$$

and

$$\exists i \in \{1, 2, \ldots, k\}, \quad f_i(\bar{x}) < f_i(\bar{x}^*) \tag{22}$$

In other words, $\bar{x}^*$ is Pareto optimal if there exists no feasible vector $\bar{x}$ which causes a reduction on some criterion without a simultaneous increase in at least another. In this context, two other notions viz., *weakly non-dominated* and *strongly non-dominated* solutions are defined [12]. A point $\bar{x}^*$ is a weakly non-dominated

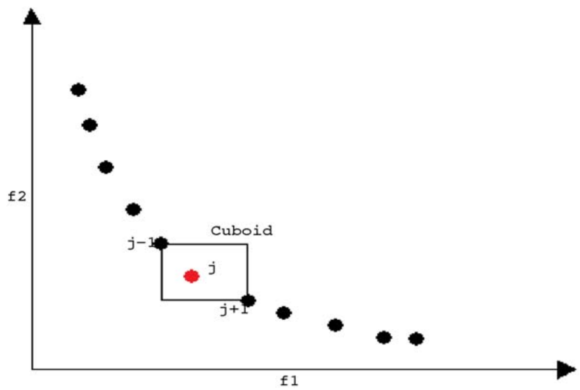

Fig. 2. The crowding distance calculation is shown.

solution if there exists no $\bar{x}$ such that $f_i(\bar{x}) < f_i(\bar{x}^*)$, for $i = 1, 2, \ldots, k$. A point $\bar{x}^*$ is a strongly non-dominated solution if there exists no $\bar{x}$ such that $f_i(\bar{x}) \le f_i(\bar{x}^*)$, for $i = 1, 2, \ldots, k$, and for atleast one i, $f_i(\bar{x}) < f_i(\bar{x}^*)$. In general, Pareto optimum usually admits a set of solutions called non-dominated solutions.

5.1. Non-dominated sorting

In this approach, every solution from the population is checked with a partially filled population of dominance. To start with, the first solution from the population is kept in the set S. Thereafter, each solution p (the second solution onwards) is compared with all members of the set S one by one. If the solution p dominates any member q of S, then solution q is removed from S. These way non-members of the non-dominated front get deleted from S. Otherwise, if solution p is dominated any member of S, then solution p is ignored. If solution p is not dominated any member of S, it is entered in S. This is how the set S grows with non-dominated solutions. When all solutions of the population are checked, the remaining members of S constitute the non-dominated set. To find the other fronts, the members of S will be discounted and the above procedure is repeated. After getting the different fronts, they are assigned through different ranks. The maximum computational complexity of this approach to find the non-dominated front is $O(\log(M \times N^2))$ where M is the number of objective and N is the size of the population [19,11].

5.2. Crwoding distance

For estimating the density of the solutions surrounding a particular solution in the population, the average distance of two points on either sides of this point along each of the objectives are computed. This quantity j_{distance} serves as an estimate of the size of the largest cuboid enclosing the point j without including any other point in the population, is called *crowding distance*. In Fig. 2, the crowding distance of the j th in its front is the average side-length of the cuboid. The crowding distance computation requires sorting of the population according to each objective function value in their ascending order of magnitude. Thereafter, for each objective function, the boundary solutions are assigned as infinity distance value. All other intermediate solutions are assigned a distance value equal to the absolute difference in the functional values of two adjacent solutions. This computation is continued with other objective functions. The overall crowding distance value is calculated as the sum of individual distance values corresponding to each objective. Computational complexity of crowding distance is $O(M \times N \times \log(N))$ [19,11].

6. Proposed multiobjective modified differential evolution based fuzzy clustering

In this section, we describe the proposed multi-objective technique MOMoDEFC for evolving a set of near-Pareto-optimal non-degenerate fuzzy partition matrices. Xie-Beni index [14] and the FCM [15,16] measure (J_m) are considered as the two objective functions that must be minimized simultaneously. The technique is described below.

6.1. Vector representation and initial population

In the MOMoDE based fuzzy clustering, the vectors are made up of real numbers which represent the coordinates of the centres of the partitions. If there are K clusters in the data set, the centres encoded in a vector in the initial population are randomly chosen K distinct points from the data set.

6.2. Computation of the objectives

In MOMoDEFC the objectives associated with each vector is computed using Eq. (1) and (9). Subsequently, the centres encoded in a vector are updated as MoDEFC.

Performance of the multiobjective clustering highly depends on the choice of objectives which should be as contradictory as possible. In this paper, the XB and J_m have been chosen as the two objectives to be optimized. From Eq. (1), it can be noted that J_m calculates the global cluster variance, i.e., it considers the within cluster variance summed up over all the clusters. Thus lower value of J_m implies better clustering solution. On the other hand, the XB index in Eq. (9) is a combination of global (numerator) and local (denominator) situations. Although the numerator of XB index [σ in Eq. (7)] is similar to J_m, the denominator contains an additional term (*sep*) representing the separation between the two nearest clusters. Therefore, the XB index is minimized by minimizing σ and by maximizing *sep*. These two terms may not attain their best values for the same partitioning when the data have complex and overlapping clusters. Considering J_m and XB (or in effect σ and *sep*) will provide a set of alternate partitionings of the data. Hence it is evident that both J_m and XB indices are needed to be minimized in order to get good solutions.

6.3. Other processes

After evaluating the fitness of all vectors, it goes through mutation (described in Eq. (15)) to generate the new offspring and crossover (described in Eq. (17)) for increasing the diversity of the mutant vector. The created offspring pool combined with its parent pool in the next step for performing the non-dominated sort [19]. Thereafter the selection process has been performed basing on the lowest rank assigned by the non-dominated sort as well as least crowding distance [19]. These processes are executed for a fixed number of iterations and final non-dominated front is the result of the MOMoDEFC algorithm. The different steps of MOMoDEFC are shown in Fig. 3.

7. Experimental results

The performance of the described MOMoDEFC scheme has been compared with the multiobjective version of differential evolution and genetic algorithm based fuzzy clustering, two single objective version of the modified differential evolution based fuzzy clustering method that minimizes the J_m and XB index, respectively, the well known fuzzy C-means algorithm and hierarchical average linkage clustering [21]. In this section results are provided for two

1. Initialize the vectors of the population.
2. Evaluate objective values of each parent vector using Eqn. 1 and 9.
3. Perform non-dominated sorting for assign rank.
4. Computes the crowding distance between members of each front.
5. Set GBest and LBest (The solution vector of lowest rank with least crowding distance).

Repeat

alpha=1/(1+exp(-(1/generation)))

if(rand(0,1)<alpha)

7. modiMutation

else

7. Mutation

end if

8.Crossover

9. Evaluate objective value of each offspring vector Eqn. 1 and 9.
10. Combine parent vectors with offspring vectors to create new population.
11. Perform non-dominated sorting for assign rank.
12. Compute the crowding distance between members of each front.
13. Update LBest (The solution vector of lowest rank with least crowding distance).
14. Set GBest with LBest
15. Select the vectors of the combined population based on non-dominated lowest rank vectors of size same as population of the next generation.

Until (termination criteria are met)

Fig. 3. Algorithm of MoMODEFC.

artificial data sets and four real-life data sets, respectively. These are described below:

7.1. Artifcial data sets

Data 1: This is an overlapping two dimensional data set where the number of clusters is five [35]. It has 250 points. The value of K is chosen to be 5. The data set is shown in Fig. 4(a).

Data 2: This is also a two dimensional data set consisting of 900 points. The data set has 9 classes [24]. The data set is shown in Fig. 4(b).

7.2. Real-life data sets

Iris: This data consists of 150 patterns divided into three classes of Iris flowers namely, Setosa, Virginia and Versicolor. The data is in four dimensional space (sepal length, sepal width, petal length and petal width).

Wine: The data set has a 178 data points along with 13 attributes (Alcohol, Malic acid, Ash, Alcalinity of ash, Magnesium, Total phenols, Flavanoids, Nonflavanoid phenols, Proanthocyanins, Color intensity, Hue, OD280/OD315 of diluted wines, Proline) and it divided into 3 classes.

Glass: It has a 214 data points which divided into 6 classes. Data points are formed on the features of Id number, Rrefractive index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron.

Cancer: It has 683 patterns in nine features (clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses), and two classes malignant and benign. The two classes are known to be linearly inseparable.

The real life data sets mentioned above were obtained from the UCI Machine Learning Repository.[1]

7.3. Input parameters

The population size and number of generation used for all the differential evolution based fuzzy clustering algorithms are 50 and 100, respectively. The crossover probability for MOMoDEFC, MODEFC, MoDEFC(J_m) and MoDEFC(XB) is taken as 0.8. The mutation factors (F and α) for all the differential algorithms are set to 0.8 and $1/(1+\exp(-(1/\text{generation})))$, respectively. MOGAFC has also executed with same number of population size and number of generation used for the differential evolution technique. The crossover and mutation probabilities for MOGAFC are taken to be 0.8 and 0.3, respectively. The FCM algorithm is executed till it converges to the final solution for different number of clusters. The FCM is executed till it converges to the final solution. For all the fuzzy clustering algorithms m, the fuzzy exponent, is set to 2.0. Results reported in the tables are the average values obtained over 50 runs of the algorithms. The results provided corresponding to the proposed multiobjective scheme is obtained by selecting the solution of the final non-dominated front that provides the best Minkowski score (MS) [30] value. For all the fuzzy clustering algorithms m, the fuzzy exponent, is set to 2.0.

7.4. Performances

Table 1 shows the comparative results obtained for the six data sets. It is seen from the table that the proposed method consistently outperforms the MODEFC, MOGAFC, single objective versions, FCM

[1] http://www.ics.uci.edu/~mlearn/MLRepository.html.

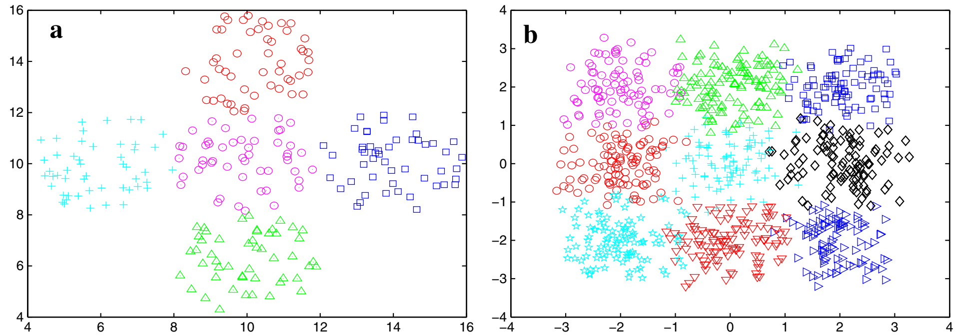

Fig. 4. Two artificial data sets (a) and (b).

as well as the average linkage algorithm in terms of the MS and ARI scores. It is also evident from the table, that individually neither the XB index (which the single objective approach optimizes) nor J_m (which the FCM optimizes) is sufficient for proper clustering. For example, for Data 1, the proposed multiobjective technique achieves an XB value of 0.3318 while the MODEFC, MOGAFC and the two single objective versions provide values of 0.3443, 0.3422, 0.4661 and 0.2337, respectively. However, when compared, in terms of the MS score and ARI value, that reflect the similarity to the true clustering, the former is found to perform much better. Similarly for J_m values, as FCM minimizes the J_m metric directly, it produces a J_m value 330.7510, whereas the proposed technique gives a value 331.6305. However, MS and ARI scores are again much better for multiobjective technique as compared to FCM. It is also observed that the value of $\Im$ index produced by MOMoDEFC is the best among all the algorithms. Similar results are found for the other data sets. Fig. 5 provides the range of solutions obtained by the different algorithms. Fig. 6 demonstrates the non-dominating Pareto front produced by MOMoDEFC, MoDEFC and MOGAFC algorithms for all the data sets. Clearly the non-dominating Pareto front produces by MOMoDEFC is better from all the data sets.

Table 1
Average of MS, ARI, J_m, XB and $\Im$ index values for the different data sets over 50 consecutive runs of different algorithms.

Data set	Algorithm	MS	ARI	J_m	XB	$\Im$
Data 1	MOMoDEFC	0.1603	0.9263	331.6305	0.3318	13.3327
	MODEFC	0.1807	0.9081	331.1421	0.3443	11.4361
	MOGAFC	0.1702	0.9107	331.5001	0.3422	12.7051
	MoDEFC(J_m)	0.2432	0.8571	326.2182	0.4661	8.0152
	MoDEFC(XB)	0.2231	0.8772	438.7063	0.2337	9.7124
	FCM	0.3616	0.8363	330.7510	0.4457	4.6103
	AL	0.4361	0.7225	450.8215	1.6210	2.3245
Data 2	MOMoDEFC	0.2661	0.8977	255.4346	0.1576	7.7631
	MODEFC	0.2751	0.8831	256.6281	0.1609	6.7341
	MOGAFC	0.2862	0.8602	258.0901	0.1459	4.0041
	MoDEFC(J_m)	0.3001	0.8771	248.9152	0.3922	5.9735
	MoDEFC(XB)	0.3212	0.8613	265.6601	0.1221	4.6623
	FCM	0.5223	0.7211	253.3215	0.3689	0.9626
	AL	0.6516	0.6105	389.0329	2.0983	0.2810
Iris	MOMoDEFC	0.2636	0.9342	62.2102	0.1274	24.3341
	MODEFC	0.2755	0.9105	65.4762	0.1121	22.5751
	MOGAFC	0.2701	0.9201	64.9762	0.1371	23.7309
	MoDEFC(J_m)	0.2935	0.9083	56.4452	0.3652	20.8741
	MoDEFC(XB)	0.3129	0.8861	71.5513	0.1021	18.6216
	FCM	0.4603	0.7832	60.8520	0.3302	5.1762
	AL	0.5962	0.6943	186.0901	2.3961	2.4932
Wine	MOMoDEFC	0.3439	0.8602	1,836,105.27	1.8661	304,545.14
	MODEFC	0.3815	0.8473	1,876,225.61	1.9407	304,430.42
	MOGAFC	0.3527	0.8577	1,887,026.61	1.6057	304,492.73
	MoDEFC(J_m)	0.4583	0.7541	1,796,007.04	2.2571	304,117.64
	MoDEFC(XB)	0.4231	0.7716	1,896,322.01	0.8010	304,204.07
	FCM	0.7329	0.5321	1,796,084.45	1.7233	303,877.61
	AL	0.9116	0.4029	1,896,402.63	3.6320	303,863.52
Glass	MOMoDEFC	0.2646	0.9265	25,308.29	1.4767	15,172.28
	MODEFC	0.2908	0.9101	22,387.82	1.7371	15,073.62
	MOGAFC	0.3192	0.8812	28,387.82	1.6371	15,007.04
	MoDEFC(J_m)	0.3602	0.8462	16,144.61	2.1261	14,864.55
	MoDEFC(XB)	0.3291	0.8754	29,335.66	0.9225	14,986.03
	FCM	0.5962	0.7372	16,204.43	1.7972	14,708.69
	AL	0.7449	0.5203	30,787.91	2.1010	14,694.72
Cancer	MOMoDEFC	0.0863	0.9775	15,105.52	1.5127	124.6332
	MODEFC	0.0962	0.9662	14,820.72	1.5562	122.9715
	MOGAFC	0.0902	0.9701	14,907.94	1.5652	123.0217
	MoDEFC(J_m)	0.1105	0.9601	14,620.06	1.6831	119.0764
	MoDEFC(XB)	0.1231	0.9462	15,186.63	0.6101	118.5291
	FCM	0.3214	0.8676	14,805.42	1.4526	65.7321
	AL	0.4445	0.7563	15,544.95	1.6792	51.0291

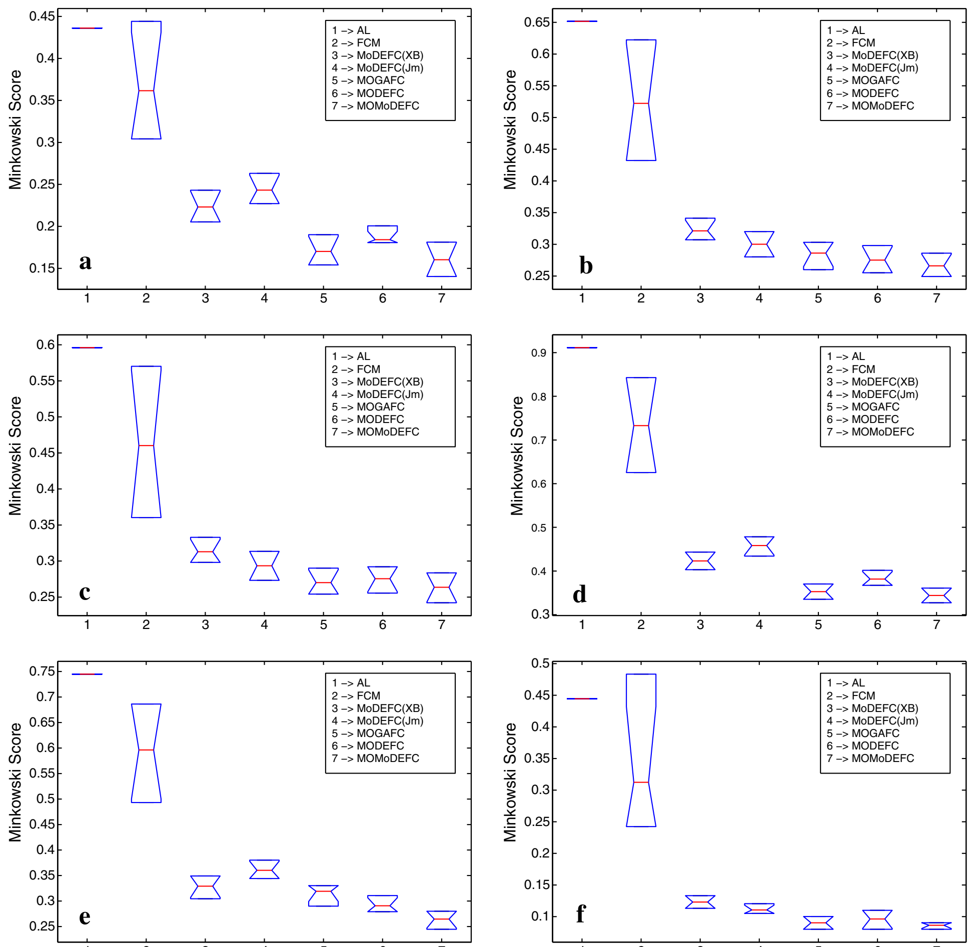

Fig. 5. Boxplot of MS for different clustering algorithms on (a) Data 1, (b) Data 2, (c) Iris, (d) Wine, (e) Glass and (f) Cancer.

7.5. Statistical significance test

A non-parametric statistical significance test called Wilcoxon's rank sum test for independent samples [36] has been conducted at the 5% significance level. Seven groups, corresponding to the seven algorithms (1. MOMoDEFC, 2. MODEFC, 3. MOGAFC, 4. MoDEFC(J_m), 5. MoDEFC(XB), 6. FCM, 7. AL), have been created for each data set. Each group consists of the Minkowski scores (MS) for the data sets produced by 50 consecutive runs of the corresponding algorithm. The median values of each group for all the data sets are shown in Table 2.

It is evident from Table 2 that the median values for MOMoDEFC are better than that for other algorithms. To establish that this goodness is statistically significant, Table 3 reports the P-values produced by Wilcoxon's rank sum test for comparison of two groups (one group corresponding to MOMoDEFC and another group corresponding to some other algorithm) at a time. As a null hypothesis, it is assumed that there are no significant differences between the median values of two groups. Whereas, the alternative hypothesis is that there is significant difference in the median values of the two groups. All the P-values reported in the table are less than 0.05 (5% significance level). For example, the rank sum test between the algorithms MOMoDEFC and MoDEFC for Data 1 provides a P-value of 0.0012, which is very small. This is strong evidence against the null hypothesis, indicating that the better median values of the performance metrics produced by MOMoDEFC is statistically significant and has not occurred by chance. Similar results are obtained for all other data sets and for all other algorithms compared to MOMoDEFC, establishing the significant superiority of the proposed technique.

8. Application for pixel classification

Two image data sets used for the experiments are two Indian remote sensing satellite (IRS) [37] images of the parts of the cities of Calcutta and Bombay [38]. Each image is of size 512×512, i.e., the size of the data set to be clustered for both images is 262,144. The IRS images consist of four bands viz., blue, green, red and near infrared. To show the effectiveness of the proposed MOMoDEFC technique quantitatively, J_m, XB and $\Im$ have been used. The efficiency of MOMoDEFC clustering can also be verified visually from the clustered images.

8.1. IRS image of Calcutta

Fig. 7a shows the Calcutta image in the NIR band. The river Hooghly cuts across the middle of the image. Towards the south-east portion, several fisheries and a township, Salt Lake, are noticed. This township is bounded on the top by a canal. Two parallel lines observed towards the north-east are the airstrips in the Dumdum airport. Moreover, there are several water bodies, roads, etc. in the image. It has been shown in [38] that this image can be well clustered into four classes which correspond to the classes turbid water (TW), pond water (PW), concrete (Concr.) and open space (OS).

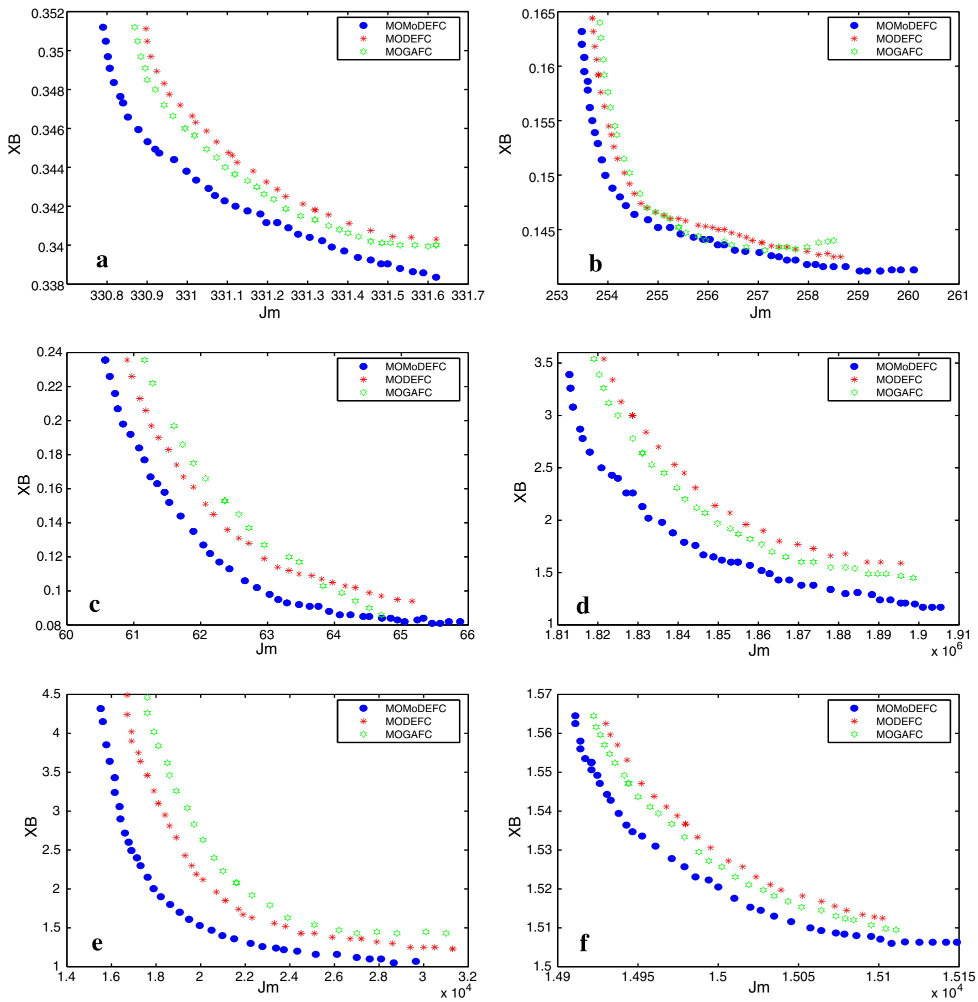

Fig. 6. Non-dominating Pareto front of MOMoDEFC, MODEFC and MOGAFC for (a) Data 1, (b) Data 2, (c) Iris, (d) Wine, (e) Glass and (f) Cancer.

Table 2
Median values of the Minkowski scores for the different data sets over 50 consecutive runs of different algorithms.

Algorithm	Data 1	Data 2	Iris	Wine	Glass	Cancer
MOMoDEFC	0.1875	0.2644	0.2606	0.3281	0.2753	0.0681
MODEFC	0.1961	0.2853	0.2762	0.3905	0.3006	0.1152
MODEFC	0.1875	0.2902	0.2688	0.3407	0.3209	0.0811
MoDEFC(J_m)	0.2537	0.3204	0.3055	0.4403	0.3602	0.1105
MoDEFC(XB)	0.2401	0.3212	0.3129	0.4231	0.3468	0.1231
FCM	0.3616	0.5223	0.4603	0.7235	0.5962	0.3124
AL	0.4361	0.6516	0.5962	0.9116	0.7449	0.4445

Fig. 7b–d shows the Calcutta image clustered using MOMoDEFC, MOGAFC and the FCM, respectively. From Fig. 7.b, it appears that the water class has been differentiated into turbid water (the river Hooghly) and pond water (canal, fisheries, etc.) as they differ in their spectral properties. Salt Lake township has come out partially as combination of concrete and open space, which appears to be correct, since this region is known to have several open spaces. The canal bounding Salt Lake has also been correctly classified as PW. Moreover, the airstrips of Dumdum airport is classified rightly as belonging to the class concrete. Presence of some small PW areas beside the airstrips is also correct as these correspond to the several ponds around the region. The concrete regions on both sides of the river, particularly towards the bottom of the image is also correct. This region refers to the central part of Calcutta.

From Fig. 7c, which shows the clustered Calcutta image using MOGAFC only, similar cluster structure is noticed as in Fig. 7b. However, careful observation reveals that MOGAFC seems to have some confusion regarding the classes concrete and PW (towards the bottom of Fig. 7c that represents the city area). It appears from Fig. 7d that the river Hooghly and the city region has been incorrectly clas-

Table 3
P-values produced by Wilcoxon's rank sum test comparing with other algorithms.

Data set	*P*-value					
	MODEFC	MOGAFC	MoDEFC(J_m)	MoDEFC(XB)	FCM	AL
Data 1	0.0012	0.0008	0.0066	0.0051	1.6327e−004	5.3075e−005
Data 2	0.0031	0.0042	0.0072	0.0063	1.3253e−004	4.9177e−005
Iris	0.0016	0.0010	0.0063	0.0068	1.3253e−004	4.8177e−005
Wine	0.0021	0.0013	0.0084	0.0072	4.4099e−004	4.9177e−005
Glass	0.0024	0.0036	0.0094	0.0088	1.4375e−004	4.7314e−005
Cancer	0.0011	0.0006	0.0075	0.0061	1.3586e−004	4.9177e−005

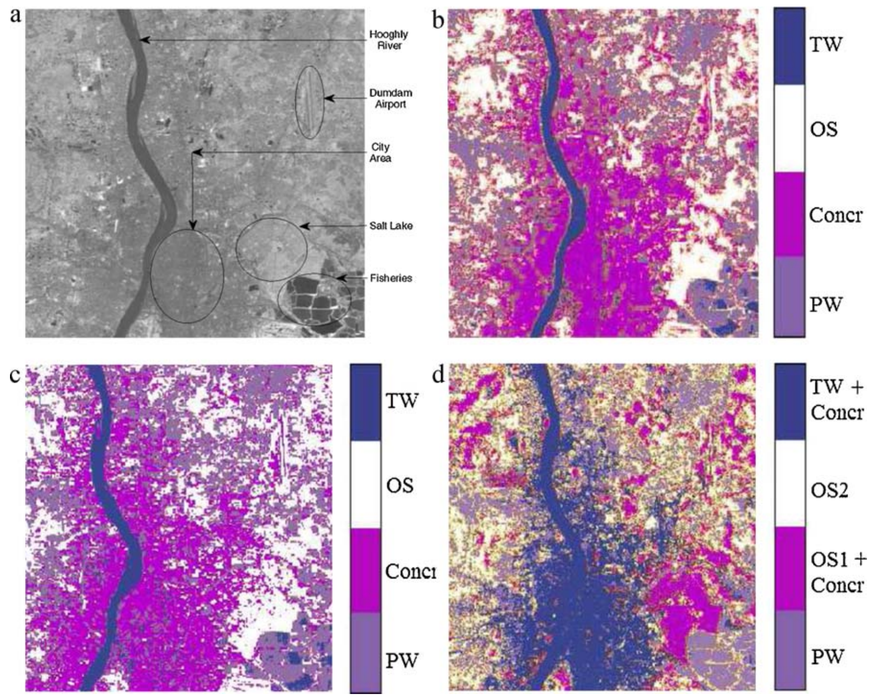

Fig. 7. (a) IRS image of Calcutta in the NIR band with histogram equalization. (b) Clustered IRS image of Calcutta using MOMoDEFC. (c) Clustered IRS image of Calcutta using MOGAFC. (d) Clustered IRS image of Calcutta using FCM.

sified as belonging to the same class by FCM. Also the whole of Salt Lake city has wrongly been put into one class. Hence the FCM result involves in a significant amount of confusion. As can be seen from Table 4, the $\Im$ index scores produced by MOMoDEFC are better than the other algorithms. This indicates that MOMoDEFC outperforms all other algorithms in terms of $\Im$ index scores.

8.2. IRS image of Bombay

Fig. 8a shows the IRS image of Bombay in the NIR band. As can be seen, the city area is surrounded on three sides by the Arabian sea. Towards the bottom right of the image, there are several islands, including the well known Elephanta islands. The dockyard is situated on the south eastern part of Bombay, which can be seen as a set of three finger like structure. This image has been classified into seven clusters namely concrete, open spaces (OS1 and OS2), vegetation (Veg), habitation (Hab), and turbid water (TW1 and TW2) [38].

The result of the application of MOMoDEFC on the Bombay image is presented in Fig. 8b. According to the available ground knowledge about the area and by visual inspection, the different clusters are labelled as, concrete (Concr.), open spaces (OS1 and OS2), vegetation (Veg), habitation (Hab) and turbid water (TW1 and TW2). The classes concrete and habitation have some common properties as habitation means low density ofconcrete. As is evident from Fig. 8b, the Arabian sea is differentiated into two classes (named as TW1 and TW2) due to different spectral properties as evident from Fig. 8a. The islands, dockyard, several road structures have mostly been correctly identified. A high proportion of open space and vegetation is noticed within the islands. The southern part of the city, which is heavily industrialized, has been classified as primarily belonging to habitation and concrete.

Fig. 8c shows the clustered Bombay image using MOGAFC clustering. Though visually it provides similar clustering structure as that provided by MOMoDEFC. However, the dockyard is not so clear as compared to the MOMoDEFC method and there are more overlaps between the two turbid water classes. Fig. 8d illustrates the Bombay image clustered using the FCM technique. It appears from the figure that the water of the Arabian sea has been partitioned into three regions, rather than two as obtained earlier. The other regions appear to be classified more or less correctly. As is evident from Table 5, for this image, MOMoDEFC provides the highest value of $\Im$ index. This indicates MOMoDEFC outperforms the other algorithms for this image.

8.3. Statistical significance test

Table 6 reports the median values of the $\Im$ index scores produced by different algorithms over 20 consecutive runs for Calcutta and Bombay images. It is evident from the table that for both the images, proposed multiobjective clustering technique produces the best median $\Im$ index scores. To establish that the better $\Im$ index

Table 4
Average J_m, XB and $\Im$ index values for IRS Calcutta image for different algorithms over 20 runs.

Algorithm	J_m	XB	$\Im$
MOMoDEFC	3,652,543.04	1.6322	110.3621
MODEFC	3,652,665.82	2.0825	103.9625
MOGAFC	3,652,651.02	1.8255	107.0405
MoDEFC(J_m)	3,651,962.06	2.7742	73.8752
MoDEFC(XB)	3,652,886.73	1.2202	66.0941
FCM	3,652,081.35	2.1409	18.0436

Table 5
Average J_m, XB and $\Im$ indices values for IRS Bombay image for different algorithms over 20 runs.

Algorithm	J_m	XB	$\Im$
MOMoDEFC	1,994,923.63	1.8342	365.6329
MODEFC	1,995,086.73	2.5782	307.0204
MOGAFC	1,994,958.53	2.1004	335.7592
MoDEFC(J_m)	1,993,005.08	5.4291	289.0642
MoDEFC(XB)	1,996,331.42	1.0921	297.1170
FCM	1,993,135.04	4.7526	85.6529

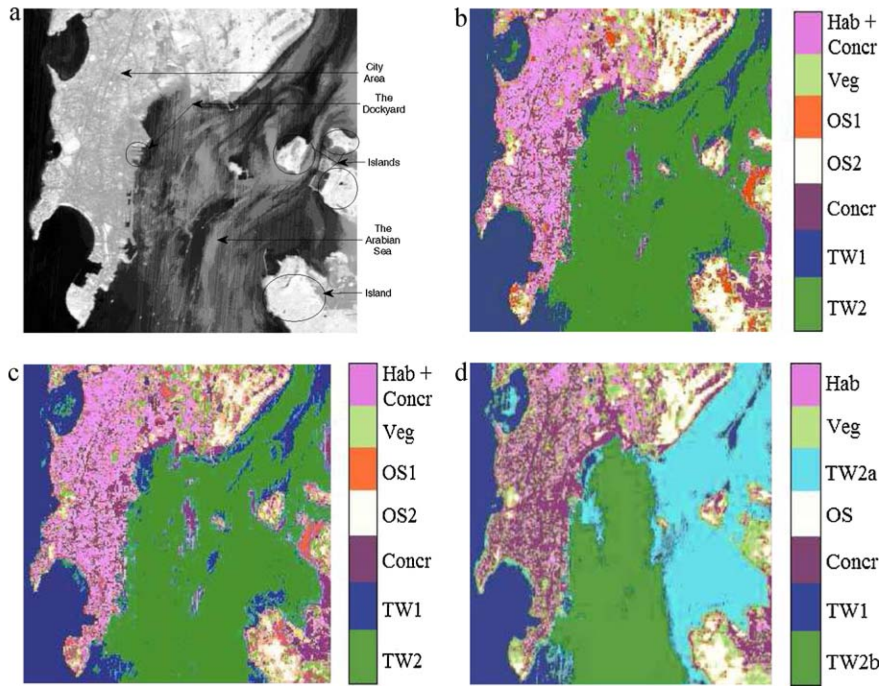

Fig. 8. (a) IRS image of Bombay in the NIR band with histogram equalization. (b) Clustered IRS image of Bombay using MOMoDEFC. (c) Clustered IRS image of Bombay using MOGAFC. (d) Clustered IRS image of Bombay using FCM.

Table 6
Median values of the $\Im$ index for the IRS Calcutta and Bombay images over 20 consecutive runs of different algorithms.

Algorithm	Calcutta	Bombay
MOMoDEFC	115.6301	361.0209
MODEFC	110.5052	315.7325
MOGAFC	112.6604	340.0704
MoDEFC(J_m)	75.6243	295.5503
MoDEFC(XB)	68.0703	293.5402
FCM	24.0962	92.9081

scores provided by proposed multiobjective technique is statistically significant and does not come by chance, it is necessary to conduct a statistical significance test.

As stated earlier, statistical significance test called Wilcoxon's rank sum has been carried out at the 5% significance level, to compare the average $\Im$ index scores produced by different algorithms. Six groups, corresponding to the six algorithms (1. MOMoDEFC, 2. MODEFC, 3. MOGAFC, 4. MoDEFC(J_m), 5. MoDEFC(XB), 6. FCM), have been created for each of the two image data considered here. Each group consists of $\Im$ index scores produced by 20 consecutive runs of the corresponding algorithm. Two groups are compared at a time, one corresponding to the MOMoDEFC scheme and the other corresponding to some other algorithm considered in this article.

Table 7 reports the results of the Wilcoxon's rank sum for the IRS Calcutta image and IRS Bombay image. The null hypothesis (The median of two groups are equal) and the alternative hypothesis is that the median of the first group is larger than the median of the second group. Also the probability (P-value) of accepting the null hypothesis is shown in the table. It is clear from the table that the P-values are much less than 0.05 (5% significance level) which are strong evidences for rejecting the null hypothesis. This proves that the better average $\Im$ index values produced by the MOMoDEFC scheme is statistically significant and has not come by chance.

Table 7
P-values produced by Wilcoxon's rank Sum test comparing with other algorithms.

Data set	*P-value*				
	MODEFC	MOGAFC	MoDEFC(J_m)	MoDEFC(XB)	FCM
Calcutta	0.0092	0.0032	0.00053	0.00072	8.7642e−006
Bombay	0.0083	0.0027	0.00084	0.00063	9.3245e−006

9. Conclusion

In this article a new multiobjective differential evolution based fuzzy clustering technique has been developed. The problem of fuzzy clustering has been modelled as optimization of cluster validity measure, namely, XB and the J_m indices. The performance of the proposed technique has been compared with multiobjective differential evolution based fuzzy clustering, multiobjective genetic algorithm based fuzzy clustering, single objective versions of differential evolution based fuzzy clustering as well as well known fuzzy C-means algorithm. A statistical significant test has been performed to establish the effectiveness of developed multiobjective technique. In this context, IRS satellite images of Calcutta and Bombay have been classified using the developed multiobjective technique and compared with other clustering algorithms. Good performance of MOMoDEFC method for large image data sets shows that it may be motivating to use this algorithm in data mining applications also.

As a scope of further research, the concept of point symmetry based clustering as proposed in [39,40] can be integrated by MOMoDEFC. The MOMoDEFC technique with other cluster validity indexes needs to be studied. Moreover, the application of MOMoDEFC algorithm may also be investigated in other real life domain including micro array data clustering [41] may also be investigated. The authors are currently working in these directions.

Acknowledgment

The authors acknowledge the reviewers whose valuable comments helped immensely in improving the quality of the article.

This work was also supported by the Polish Ministry of Education and Science (grants N301 159735, N518 409238, and others).

References

[1] J.A. Hartigan, Clustering Algorithms, Wiley, 1975.

[2] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[3] B.S. Everitt, Cluster Analysis, third ed., Halsted Press, 1993.

[4] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data, Wiley, 2005.

[5] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces, Technical Report TR-95-012, 1995.

[6] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic strategy for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.

[7] K. Price, R. Storn, J. Lampinen, Differential Evolution – A Practical Approach to Global Optimization, Springer, Berlin, 2005.

[8] U. Maulik, S. Bandyopadhyay, I. Saha, Integrating clustering and supervised learning for categorical data analysis, IEEE Transactions on Systems, Man and Cybernetics Part A (2010), doi:10.1109/TSMCA.2010.2041225.

[9] U. Maulik, I. Saha, Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery, Pattern Recognition 42 (9) (2009) 2135–2149.

[10] U. Maulik, I. Saha, Automatic fuzzy clustering using modified differential evolution for image classification, IEEE Transactions on Geoscience and Remote Sensing (2010), doi:10.1109/TGRS.2010.2047020.

[11] K. Deb, Multi-objective Optimization Using Evolutionary Algorithms, John Wiley and Sons, Ltd., England, 2001.

[12] C.A. Coello Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowledge and Information Systems 1 (3) (1999) 129–156.

[13] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Tech. Rep. 103, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.

[14] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 841–847.

[15] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.

[16] N.R. Pal, J.C. Bezdek, On cluster validity for the Fuzzy C-Means model, IEEE Transactions on Fuzzy Systems 3 (1995) 370–379.

[17] S. Bandyopadhyay, Simulated annealing using a reversible jump markov chain monte carlo algorithm for fuzzy clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 479–490.

[18] I. Saha, Use of multiobjective differential fuzzy clustering with ann classifier for unsupervised pattern classification: application to microarray analysis, in: Proceedings of 4th Int. Conf. on Research Challenges in Information Science, RCIS'10, May, 2010, pp. 651–659.

[19] K. Deb, S. Agrawal, A. Pratab, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.

[20] S. Bandyopadhyay, U. Maulik, A. Mukhopadhyay, Multiobjective genetic clustering for pixel classification in remote sensing imagery, IEEE Transactions on Geoscience and Remote Sensing 45 (5) (2007) 1506–1511.

[21] J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974.

[22] M. Delattre, P. Hansen, Bicriterion cluster analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence vol.2 (4) (1980) 277–291.

[23] R. Caballero, M. Laguna, R. Marti, J. Molina, Multiobjective clustering with metaheuristic optimization technology, tech. rep., Leeds School of Business in the University of Colorado at Boulder, CO, 2006.

[24] U. Maulik, S. Bandyopadhyay, Genetic algorithm based clustering technique, Pattern Recognition 33 (2000) 1455–1465.

[25] J. Handl, J. Knowles, Multiobjective clustering around medoids, in: Proceedings of IEEE Congress on Evolutionary Computation, vol. 1, 2005, pp. 632–639.

[26] J. Handl, J. Knowles, An evolutionary approach to multiobjective clustering, IEEE Transactions on Evolutionary Computation 11 (2006) 56–76.

[27] J. Handl, J. Knowles, Multiobjective clustering and cluster validation, Computational Intelligence 16 (2006) 21–47.

[28] S. Bandyopadhyay, U. Maulik, An evolutionary technique based on k-means algorithm for optimal clustering in r^n, Information Science 146 (2002) 221–237.

[29] S. Bandyopadhyay, An efficient technique for superfamily classification of amino acid sequences: feature extraction, fuzzy clustering and prototype selection, Fuzzy Sets and Systems 152 (2005) 5–16.

[30] N. Jardine, R. Sibson, Mathematical Taxonomy, John Wiley and Sons, 1971.

[31] K.Y. Yeung, W.L. Ruzzo, An empirical study on principal component analysis for clustering gene expression data, Bioinformatics 17 (2001) 763–774.

[32] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics 28 (1998) 301–315.

[33] L. Hubert, P. Arabie, Some new indexes of cluster validity, Journal of Classification 2 (1985) 193–218.

[34] U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1650–1654.

[35] A. Mukhopadhyay, U. Maulik, Unsupervised pixel classification in satellite imagery: a two-stage fuzzy clustering approach, Fundamenta Informaticae (2008) 1001–1018.

[36] M. Hollander, D.A. Wolfe, Nonparametric Statistical Methods, 2nd ed., 1999.

[37] IRS Data Users Handbook, NRSA, Hyderabad, India, 1986.

[38] U. Maulik, S. Bandyopadhyay, Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification, IEEE Transactions on Geoscience and Remote Sensing 41 (5) (2003) 1075–1081.

[39] S. Bandyopadhyay, S. Saha, GAPS: a new symmetry based genetic clustering technique, Pattern Recognition 40 (2007) 3430–3451.

[40] S. Bandyopadhyay, S. Saha, A point symmetry based clustering technique for automatic evolution of clusters, IEEE Transactions on Knowledge and Data Engineering 20 (11) (2008) 1–17.

[41] S. Bandyopadhyay, A. Mukhopadhyay, U. Maulik, An improved algorithm for clustering gene expression data Bioionformatics, vol. 23, no. 21, Oxford University Press, 2007, pp. 2859–2865.