

# Automatic Clustering with Multi-objective Differential Evolution Algorithms

Kaushik Suresh<sup>1</sup>, Debarati Kundu<sup>1</sup>, Sayan Ghosh<sup>1</sup>, Swagatam Das<sup>1</sup> and Ajith Abraham<sup>2,3</sup>

<sup>1</sup>Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata, India  
swagatamdas19@yahoo.co.in

<sup>2</sup>Center of Excellence for Quantifiable Quality of Service, Norwegian University of Science and Technology, Norway

<sup>3</sup>Machine Intelligence Research Labs - MIR Labs  
ajith.abraham@ieee.org

**Abstract**—This paper applies the Differential Evolution (DE) algorithm to the task of automatic fuzzy clustering in a Multi-objective Optimization (MO) framework. It compares the performances of four recently developed multi-objective variants of DE over the fuzzy clustering problem, where two conflicting fuzzy validity indices are simultaneously optimized. The resultant Pareto optimal set of solutions from each algorithm consists of a number of non-dominated solutions, from which the user can choose the most promising ones according to the problem specifications. A real-coded representation of the search variables, accommodating variable number of cluster centers, is used for DE. The performances of four DE variants have also been contrasted to that of two most well-known schemes of MO clustering namely the Non Dominated Sorting Genetic Algorithm (NSGA II) and Multi-Objective Clustering with an unknown number of Clusters K (MOCK). Experimental results over six artificial and four real life datasets of varying range of complexities indicates that DE holds immense promise as a candidate algorithm for devising MO clustering schemes.

## I. INTRODUCTION

Optimization-based automatic clustering algorithms greatly rely on a cluster validity function (optimization criterion) the optima of which appear as proxies for the unknown “correct classification” in a previously unhandled dataset [1]. Different formulations of the clustering problem vary in the optimization criterion used. Most existing clustering methods, however, attempt to optimize just one such clustering criterion modeled by a single cluster validity index. This often results into considerable discrepancies observable between the solutions produced by different algorithms on the same data. A single cluster validity measure is hardly able to judge the correctness of clustering for a wide variety of real life datasets. A wrong choice of the validity measure may lead to poor clustering results. Thus, the single-objective clustering method may prove futile (as judged by means of expert’s knowledge) in a context where the criterion employed is inappropriate. In situations where the best solution corresponds to a tradeoff between

different conflicting objectives, common sense advocates a multi-objective framework for clustering. In the case of iterative optimization algorithms, it is possible that a single-objective approach would visit such tradeoff solutions during a run, but would not recognize them as good and discard them.

Although there has been a plethora of papers reporting several single-objective evolutionary clustering techniques (a comprehensive survey of which can be found in [1, 2]), very few research works have so far been undertaken towards the application of evolutionary multi-objective optimization algorithms (EMOA) for pattern clustering [3, 4]. A state-of-the-art literature survey indicates that DE has already proved itself as a promising candidate in the field of evolutionary multi-objective optimization (EMO) [5 – 8]. Earlier it has also been successfully applied to single-objective partitional clustering [9 – 11].

The work reported in [3] is based on Deb *et al.*’s celebrated NSGA (Non Dominated Sorting genetic Algorithm)-II [12] and the clustering method described in [4] is based on PESA (Pareto Evolution based Selection) II [13], and both the algorithms are multi-objective variants of Genetic Algorithm (GA). However, the multi-objective variants of DE have not been applied to the general data clustering problems till date, to the best of our knowledge. This paper primarily compares the performances of four most representative multi-objective DE algorithms on the multi-objective fuzzy clustering problem. The multi-objective DE-variants considered here are namely the Pareto DE (PDE) [5], the Multi-objective DE (MODE) [6], DE for Multi-objective Optimization (DEMO) [7], and Non-Dominated Sorting DE (NSDE) [8]. Since DE, by nature, is a real-coded population-based optimization algorithm, we here resort to centroid-based representation scheme for the search variables. Note that in contrast to single objective optimization that yields a single best solution, in MOO, a number of, often conflicting, objective functions are optimized simultaneously and thus an MOO algorithm, in general, ends up with a number of Pareto optimal solutions. None of these Pareto optimal solutions can be improved upon an objective any further without degrading it on another. Here we consider the Xie-Beni index [14] and the Fuzzy

C Means (FCM) measure ( $J_m$ ) [15] as the objective functions. Note that any other and any number of objective functions could be used in the proposed MOO clustering framework. The performance of the multi-objective DE-variants have also been contrasted with two best-known EMOA-based clustering methods till date. The first one of these is MOCK by Handl and Knowles [4] while the second one is based on NSGA II and was used by Bandyopadhyay *et al.* for pixel clustering in remote sensing satellite image data [3]. Although we experimented with a large variety of datasets, here we report the results for ten representative datasets including the microarray Yeast sporulation data [16].

## II. MULTI-OBJECTIVE OPTIMIZATION WITH DE

### 2.1 The MO Problem

In many practical or real life problems, there are many (possibly conflicting) objectives that need to be optimized simultaneously. Under such circumstances there no longer exists a single optimal solution but rather a whole set of possible solutions of equivalent quality. The field of Multi-objective Optimization (MO) [17 – 19] deals with simultaneous optimization of multiple, possibly competing, objective functions. The MO problems tend to be characterized by a family of alternatives, which must be considered equivalent in the absence of information concerning the relevance of each objective relative to the others.

The family of solutions of an MO problem is composed of the parameter vectors, which cannot be improved in any objective without causing degradation in at least one of the other objectives. This forms the central idea of *Pareto-optimality*. The concepts of *dominance* and *Pareto-optimality* may be presented more formally in the following way [18, 19]:

**Definition 1:** Consider without loss of generality the following multi-objective optimization problem with  $m$  decision variables  $x$  (parameters) and  $n$  objectives :

Maximize:

$$\vec{Y} = f(\vec{X}) = (f_1(x_1, \dots, x_m), \dots, f_n(x_1, \dots, x_m)), \quad (1)$$

where  $\vec{X} = [x_1, \dots, x_m]^T \in P$  and  $\vec{Y} = [y_1, \dots, y_n]^T \in O$  and

where  $\vec{X}$  is called decision (parameter) vector,  $P$  is the parameter space,  $\vec{Y}$  is the objective vector, and  $O$  is the objective space. A decision vector  $\vec{A} \in P$  is said to dominate another decision vector  $\vec{B} \in P$  (also written as  $\vec{A} \succ \vec{B}$ ) if and only if:

$$\forall i \in \{1, \dots, n\} : f_i(\vec{A}) \geq f_i(\vec{B})$$

$$\wedge \exists j \in \{1, \dots, n\} : f_j(\vec{A}) > f_j(\vec{B}) \quad (2)$$

Based on this convention, we can define non-dominated, *Pareto-optimal* solutions as follows:

**Definition 2:** Let  $\vec{A} \in P$  be an arbitrary decision vector.

(a) The decision vector  $\vec{A}$  is said to be non-dominated regarding the set  $P' \subseteq P$  if and only if there is no vector in  $P'$  which can dominate  $\vec{A}$ . Formally,

$$\nexists \vec{A}' \in P' : \vec{A}' \succ \vec{A} \quad (3)$$

(b) The decision (parameter) vector  $\vec{A}$  is called Pareto-optimal if and only if  $\vec{A}$  is non-dominated regarding the whole parameter space  $P$ .

### 2.2 The Differential Evolution (DE) Algorithm

DE [20, 21] is a population-based global optimization algorithm that uses a floating-point (real-coded) representation. The  $i$ -th individual (parameter vector or *chromosome*) of the population at generation (time)  $G$  is a  $D$ -dimensional vector containing a set of  $D$  optimization parameters:

$$\vec{Z}_{i,G} = [Z_{i,1,G}, Z_{i,2,G}, \dots, Z_{i,D,G}] \quad (4)$$

Now, in each generation, a donor  $\vec{Y}_{i,G}$  is created. The method of creating this donor vector demarcates between the various DE schemes. In one of the earliest variants of DE, now called DE/rand/1 scheme, three other parameter vectors (say the  $r_1$ ,  $r_2$ , and  $r_3$ -th vectors such that  $r_1, r_2, r_3 \in [1, NP]$  and  $r_1 \neq r_2 \neq r_3$  are chosen at random from the current population. Next the difference of any two of the three vectors is multiplied by a scalar number  $F$  and the scaled difference is added to the third one, whence we obtain the donor vector  $\vec{Y}_{i,G}$ . The process for the  $j$ -th component of the  $i$ -th vector may be expressed as,

$$Y_{i,j,G} = Z_{r_1,j,G} + F \cdot (Z_{r_2,j,G} - Z_{r_3,j,G}) \quad (5)$$

Next a crossover operation takes place to increase the potential diversity of the population. We use ‘binomial’ crossover in which case the number of parameters inherited from the mutant has a (nearly) binomial distribution. Thus for each target vector  $\vec{Z}_{i,G}$ , a trial vector  $\vec{R}_{i,G}$  is created in the following fashion:

$$R_{j,i,G} = Y_{j,i,G}, \quad \text{if } (rand_{i,j}(0,1) \leq Cr \text{ or } j = j_{rand}) \\ Z_{j,i,G}, \quad \text{otherwise} \quad (6)$$

for  $j = 1, 2, \dots, D$  and  $rand_j(0, 1) \in [0,1]$  is the  $j$ -th evaluation of a uniform random number generator.  $j_{rand} \in [1, 2, \dots, D]$  is a randomly chosen index which ensures that  $\vec{R}_i(t)$  gets at least one component from  $\vec{Z}_i(t)$ . Next step of the algorithm calls for ‘selection’ in order to determine which one between the target vector and trial vector will survive in the next generation i.e. at time  $t = t+1$ . If the trial vector yields a better value of the fitness function, it replaces its target vector in the next

generation; otherwise the parent is retained in the population:

$$\left. \begin{aligned} \bar{Z}_i(t+1) &= \bar{R}_i(t) && \text{if } f(\bar{R}_i(t)) \leq f(\bar{Z}_i(t)) \\ &= \bar{Z}_i(t) && \text{if } f(\bar{R}_i(t)) > f(\bar{Z}_i(t)) \end{aligned} \right\} \quad (7)$$

where  $f(\cdot)$  is the function to be minimized.

### 2.3 The Multi-objective Variants of DE

We compared the performances of four recently developed and popular MO-variants of DE: the Pareto DE (PDE) [3], the Multi-objective DE (MODE) [4], DE for Multi-objective Optimization (DEMO) [5], and Non-Dominated Sorting DE (NSDE) [6]. Due to space limitations, we briefly discuss here the outline of these algorithms instead of reiterating through their details available in cited literatures.

**1) PDE:** In the PDE algorithm proposed by Abbas and Sarker, an initial population is generated at random from a Gaussian distribution with mean 0.5 and standard deviation 0.15. All dominated solutions are removed from the population. The remaining non-dominated solutions are retained for reproduction. If the number of non-dominated solutions exceeds some threshold, a distance metric relation (will be described in the next paragraph) is used to remove those parents who are very close to each others [17, 18]. Three parents are selected at random. A child is generated from the three parents and is placed into the population if it dominates the first selected parent; otherwise a new selection process takes place

**2) MODE:** MODE was proposed by Xue *et al.* in [8]. This algorithm uses a variant of the original DE, in which the best individual is adopted to create the offspring. A Pareto-based approach is introduced to implement the selection of the best individual. If a solution is dominated, a set of non-dominated individuals can be identified and the “best” turns out to be any individual (randomly picked) from this set. Also, the authors adopt  $(\mu + \lambda)$  selection, Pareto ranking and crowding distance in order to produce and maintain well-distributed solutions.

**3) DEMO:** The algorithm, proposed by Robic and Filipic combines the advantages of DE with the mechanisms of Pareto-based ranking and crowding distance sorting. DEMO only maintains one population and it is extended when newly created candidates take part immediately in the creation of the subsequent candidates. This enables a fast convergence towards the true Pareto front, while the use of non-dominated sorting and crowding distance (derived from the NSGA-II [12]) of the extended population promotes the uniform spread of solutions.

**4) NSDE:** In Iorio and Li’s NSDE algorithm, within the NSGA-II framework, the DE-variants are used to generate  $N$  offspring from the selected parents. The offspring individuals are then evaluated on the objective functions. Following this, they are combined with the parent generation. The combined population is then sorted into

dominance ranks, as was mentioned previously. Each individual also has a crowding distance associated with it. New candidates are generated using the DE/current-to-rand/1 strategy. NSDE is used to solve rotated problems with a certain degree of rotation on each plane.

## III. MULTI-OBJECTIVE CLUSTERING SCHEME

### 3.1 Search-variable Representation

In the proposed method, for  $n$  data points, each  $d$ -dimensional, and for a user-specified maximum number of clusters  $K_{\max}$ , a chromosome is a vector of real numbers of dimension  $K_{\max} + K_{\max} \times d$ . The first  $K_{\max}$  entries are positive floating-point numbers in  $[0, 1]$ , each of which controls whether the corresponding cluster is to be activated (i.e. to be really used for classifying the data) or not. The remaining entries are reserved for  $K_{\max}$  cluster centers, each  $d$ -dimensional. For example, the  $i$ -th vector is represented as:

$$\bar{X}_i(t) = \begin{array}{|c|c|c|c|c|c|c|c|} \hline T_{i,1} & T_{i,2} & \dots & T_{i,K_{\max}} & \bar{m}_{i,1} & \bar{m}_{i,2} & \dots & \bar{m}_{i,K_{\max}} \\ \hline \end{array} \quad (8)$$

The  $j$ -th cluster center in the  $i$ -th chromosome is active or selected for partitioning the associated dataset if  $T_{i,j} > 0.5$ . On the other hand, if  $T_{i,j} < 0.5$ , the particular  $j$ -th cluster is inactive in the  $i$ -th vector in DE population. Thus the  $T_{i,j}$ s behave like control genes (we call them *activation thresholds*) in the vector governing the selection of the active cluster centers. The rule for selecting the actual number of clusters specified by one vector is:

**IF**  $T_{i,j} > 0.5$  **THEN** the  $j$ -th cluster center  $\bar{m}_{i,j}$  is

**ACTIVE ELSE**  $\bar{m}_{i,j}$  is **INACTIVE**. (9)

### 3.2 Selecting the Objective Functions

The performance of a multi-objective clustering algorithm critically depends upon the clustering objectives it tries to optimize simultaneously. Conflict among the objective functions is often beneficial since it guides to globally optimal solutions. It also ensures that no single clustering objective is optimized leaving other probable significant objectives unnoticed.

In this work, we choose the Xie-Beni index  $XB_q$  and the FCM objective function  $J_q$  as the two objectives. The FCM measure  $J_q$  may be defined as:

$$J_q = \sum_{j=1}^n \sum_{i=1}^k u_{ij}^q \cdot d^2(\bar{Z}_j, \bar{m}_i), \quad 1 \leq q \leq \infty \quad (10)$$

where  $q$  is the fuzzy exponent,  $d$  indicates a distance measure between the  $j$ -th pattern vector and  $i$ -th cluster

centroid, and  $u_{ij}$  denotes the membership of  $j$ -th pattern in the  $i$ -th cluster. The XB index is defined as a function of the ratio of the total variation  $\sigma$  to the minimum separation sep of the clusters. Here  $\sigma$  and sep may be written as:

$$\sigma = \sum_{i=1}^k \sum_{p=1}^n u_{ip}^2 \cdot d(\bar{m}_i, \bar{Z}_p) \quad (11)$$

$$\text{and } \text{sep}(Z) = \min_{i \neq j} \left\{ d^2(\bar{m}_i, \bar{m}_j) \right\} \quad (12)$$

The XB index is then written as:

$$XB_q = \frac{\sigma}{n \times \text{sep}(Z)} = \frac{\sum_{i=1}^k \sum_{p=1}^n u_{ip}^q \cdot d^2(\bar{m}_i, \bar{Z}_p)}{n \times \min_{i \neq j} \left\{ d^2(\bar{Z}_i, \bar{Z}_j) \right\}} \quad (13)$$

Note that when the partitioning is compact and the individual clusters are well separated, value of  $\sigma$  should be low while sep should be high, thereby yielding lower values of  $XB_q$  index. The objective therefore is to minimize the XB index. For computing the measures described in equations (10) and (13), the centers encoded in a DE vector are first extracted. Let the set of centers be denoted by  $\{\bar{m}_1, \bar{m}_2, \dots, \bar{m}_k\}$ . The membership value of the  $j$ -th pattern in  $i$ -th cluster  $u_{ij}, i = 1, 2, \dots, k$  and  $j = 1, 2, \dots, n$  are computed as:

$$u_{ij} = \frac{1}{\sum_{p=1}^k \left( \frac{d(\bar{m}_i, \bar{Z}_j)}{d(\bar{m}_p, \bar{Z}_j)} \right)^{\frac{2}{q-1}}} \quad (14)$$

Note that while computing the  $u_{ij}$  s, using equation (12), if  $d(\bar{m}_p, \bar{Z}_j)$  is equal to zero for some  $p$ , then  $u_{ij}$  is set to zero for all  $i = 1, 2, \dots, k, i \neq j$ , while  $u_{pj}$  is set equal to one. Subsequently the centers encoded in a vector are updated using the following equation:

$$\bar{m}_p = \frac{\sum_{j=1}^n (u_{pj})^q \cdot \bar{Z}_j}{\sum_{j=1}^n (u_{pj})^q} \quad (15)$$

and the cluster membership values are recomputed. Note that the  $XB_q$  index is a combination of global (numerator) and particular (denominator) situations. The numerator is similar to  $J_m$  but the denominator has a factor that gives the separation between to minimum distant clusters. Hence this factor only considers the worst case, i.e. which two clusters are closest to each other and forgets about the other partitions. Here, greater value of the denominator (lower value of whole index) signifies a better partitioning. Thus it is evident that  $J_m$  and  $XB_q$  indices

should be simultaneously minimized in order to get good solutions. The two terms at the numerator and the denominator of  $XB_q$  may not attain their best values for the same partitioning when the data has complex and overlapping clusters, such as remote sensing image data.

### 3.3 Avoiding Erroneous Vectors

There is a possibility that in our scheme, during computation of the XB or  $J_q$ , a division by zero may be encountered. This may occur when one of the selected cluster centers in a DE-vector is outside the boundary of distributions of the data set. To avoid this problem we first check to see if any cluster has fewer than two data points in it. If so, the cluster center positions of this special chromosome are re-initialized by an average computation. We put  $n/k$  data points for every individual cluster center, such that a data point goes with a center that is nearest to it

### 3.4 Selecting the Best Solution from Pareto-front

Multi-objective clustering does not return a single solution, but a set of clustering solutions. These individual groupings correspond to different tradeoffs between the two objectives and, in our case, also consist of different numbers of clusters. Several researchers have already investigated the identification of promising solutions from Pareto front approximations recently [22, 23]. For choosing the most interesting solutions from the Pareto front, we apply Tibshirani *et al.*'s Gap statistic [24], a statistical method to determine the number of clusters in a data set. The Gap statistic is based on the expectation that the most suitable number of clusters shows in a significant "knee" when plotting the performance of a clustering algorithm (in terms of a selected internal evaluation measure) as a function of the number of clusters.

### 3.5 Evaluating the Clustering Quality

In this work, the final clustering quality is evaluated using two external measures. Specifically we choose the Adjusted Rand Index [25] (which is a generalization of the Rand index [26]) and the Silhouette index [27]. Silhouette width reflects the compactness and separation of the clusters. Given a set of data points  $Z = \{\bar{Z}_1, \dots, \bar{Z}_n\}$  and a given clustering solution  $C = \{C_1, C_2, \dots, C_k\}$ , the silhouette width  $s(\bar{Z}_j)$  for each data  $\bar{Z}_j$  belonging to cluster  $C_i$  indicates a measure of the confidence of belongingness, and it is defined as:

$$s(\bar{Z}_j) = \frac{b(\bar{Z}_j) - a(\bar{Z}_j)}{\max(a(\bar{Z}_j), b(\bar{Z}_j))}. \quad (16)$$

Here  $a(\vec{Z}_j)$  denotes the average distance of data point  $\vec{Z}_j$  from the other data points of the cluster to which the data point  $\vec{Z}_j$  is assigned (i. e. cluster  $C_i$ ). On the other hand,  $b(\vec{Z}_j)$  represents the minimum of the average distances of data point  $\vec{Z}_j$  from the data points belonging to clusters  $C_r$ ,  $r=1,2,\dots,k$  and  $r \neq i$ . The value of  $s(\vec{Z}_j)$  lies between -1 and +1. Large values of  $s(\vec{Z}_j)$  (near to 1) indicate that the data point  $\vec{Z}_j$  is well clustered. Value of  $s(\vec{Z}_j)$  around 0 means that the data point lays between two clusters and a negative value of  $s(\vec{Z}_j)$  indicates that the data point  $\vec{Z}_j$  is probably placed in a wrong cluster. Overall silhouette index  $s(C)$  of a clustering solution  $C = \{C_1, C_2, \dots, C_k\}$  is defined as the mean silhouette width over all the data points:

$$s(C) = \frac{1}{n} \sum_{j=1}^n s(\vec{Z}_j) \quad (17)$$

Greater values of  $s(C)$  (near to 1) reflect that most of the data points are correctly clustered and this in turn indicates a better clustering solution. Silhouette index can be evaluated for any distance measure.

#### IV. EXPERIMENTAL RESULTS

##### 4.4.1 Datasets used

The experimental results showing the effectiveness of multi-objective DE based clustering has been provided for six artificial and four real life datasets. The artificial datasets are named as Dataset\_1 to Dataset\_6 with number of clusters varying from 3 to 10. Table 1 presents the number of objects, dimensionality and the number of clusters for each data. The real-life datasets are iris, wine, breast-cancer [28] and the yeast sporulation data. We consider here the microarray data on the transcriptional program of sporulation in budding yeast, the collection and analysis of which have been described in [16]. The sporulation dataset is available publicly from [31]. This dataset consists of 6118 genes measured across 7 time points (0, 0.5, 2, 5, 7, 9, and 11.5 h) during the sporulation process of budding yeast. The data are then log-transformed. Among the 6118 genes, the genes, whose expression levels did not change significantly during the harvesting, have been ignored from further analysis. This is determined with a threshold level of 1.6 for the root mean squares of the log2-transformed ratios. The resulting set consists of 474 genes. Please note that for the yeast sporulation dataset, we have used the Pearson correlation coefficient based distance measure [29], instead of the conventional Euclidean distance (which has been used for the rest of the datasets), as it has been

shown to be more effective for clustering microarray datasets [29].

##### 4.2 Parameters for the Algorithms

All the multi-objective DE variants have been used with 40 parameter vectors in each generation and each run of each algorithm was continued for 100 generations. The value of scale factor  $F$  is a random value between 0.5 and 1. The other parameters for the multi-objective GA (NSGA II) based clustering are fixed as follows: number of generations = 100, population size = 50, crossover probability = 0.8, mutation probability =  $\frac{1}{\text{Chromosome\_length}}$ . Please note that the four DE variants and the NSGA II use the same parameter representation scheme.

##### 4.3 Presentation of Results

The mean Silhouette index values of the best-of-run solutions provided by six contestant algorithms over the 10 datasets have been provided in Table 2. The best entries have been marked in boldface in each row. Table 3 enlists the adjusted rand index values except for Yeast sporulation data as no standard nominal classification is known for this dataset. Table 4 shows the results of unpaired  $t$  tests (standard error of difference of the two means, 95% confidence interval of this difference, the  $t$  value, and the two-tailed P value) between the best and second best algorithms in terms of average Silhouette index (we omit a similar table for adjusted rand index due to space limitations). For all cases in Table 4, sample size = 30 and number of degrees of freedom = 58. The results listed in Tables 2 to 4 indicate that there is always one or more multi-objective DE variant that beats the NSGA II or MOCK in terms of mean Silhouette index in a statistically significant fashion.

TABLE 1. DETAILS OF THE DATASETS USED.

Dataset	Number of points	Number of clusters	Number of Characteristics
Dataset_1	900	9	2
Dataset_2	76	3	2
Dataset_3	400	4	3
Dataset_4	300	6	2
Dataset_5	500	10	2
Dataset_6	810	3	2
Iris	150	3	4
Wine	178	3	13
Breast-Cancer	683	2	9
Yeast Sporulation	474	7	7

##### 4.4 Significance and Data Clustering Results

The best clustering solution provided by different algorithms on the sporulation data of yeast has been

visualized using the cluster profile plot (in parallel coordinates) and the heatmap plot in MATLAB 7.0.4 version. Parallel coordinates [41] is a common way of visualizing high-dimensional geometry. A point in  $n$ -dimensional space is represented as a polyline with vertices on the parallel axes; the position of the vertex on the  $i$ -th axis corresponds to the  $i$ -th coordinate of the point. Cluster profile plots (in parallel coordinates) of seven clusters for the best clustering result (provided by MODE) on yeast sporulation data has been shown in Figure 1. The blue polylines indicate the member genes within a cluster while the black polyline indicates the centroid of that gene. Cluster profile plots (Figure 1) also demonstrate how the cluster profiles for the different groups of genes differ from each other, while the profiles within a group are reasonably similar.

## V. CONCLUSIONS

This paper compared and contrasted the performances of four state-of-the-art multi-objective variants of DE in an automatic clustering framework with two other prominent multi-objective clustering algorithms. The multi-objective DE-variants used the same variable representation scheme. Tables 2 to 4 indicate that one or more multi-objective DE variants were always able to produce better final clustering solutions as compared to MOCK or NSGA II when all the algorithms were let run for an equal number of generations. Visualization of the yeast sporulation data clustering results indicate that the MODE yielded compact and well separated clusters. Our experimental results indicate that DE holds immense promise as a candidate optimization technique for multi-objective clustering. Future research may extend the multi-objective DE-based clustering schemes to handle discrete chromosome representation schemes that no longer depend on cluster centroids and thus are not biased in any sense towards spherical clusters. As a scope of further research, the technique of multi-objective optimization with other cluster validity indices needs to be studied. Moreover, new ways of comparing the performance of multi-objective solutions have to be defined.

## REFERENCES

1. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review", *ACM Computing Surveys*, vol. 31, no.3, (1999) 264–323.
2. R. Xu and D. Wunsch, *Clustering*, Series on Computational Intelligence, IEEE Press, 2008.
3. S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery", *IEEE Transactions Geoscience and Remote Sensing*, 2006.
4. J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering", *IEEE Transactions on Evolutionary Computation*, 11(1):56-76, 2007.
5. H. A. Abbass and R. Sarker, "The pareto differential evolution algorithm", *International Journal on Artificial Intelligence Tools*, 11(4):531-552, 2002.
6. F. Xue, A. C. Sanderson, and R. J. Graves, "Pareto-based multi-objective differential evolution", in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 862–869, Canberra, Australia, 2003, IEEE Press.
7. T. Robic and B. Filipic, "DEMO: Differential Evolution for Multiobjective Optimization", In C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, editors, *Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005*, pages 520–533, Guanajuato, Mexico, 2005, Springer. Lecture Notes in Computer Science Vol. 3410, 2005.
8. A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution", in *AI 2004: Advances in Artificial Intelligence, Proceedings*, pages 861–872, Springer-Verlag, Lecture Notes in Artificial Intelligence Vol. 3339, 2004.
9. S. Paterlinia, T. Krink, "Differential evolution and particle swarm optimisation in partitional clustering", *Computational Statistics & Data Analysis*, Volume 50, Issue 5, 1220-1247, 2006.
10. M. Omran, A. P. Engelbrecht and A. Salman, "Differential evolution methods for unsupervised image classification", *Proceedings of Seventh Congress on Evolutionary Computation (CEC-2005)*. IEEE Press, 2005.
11. S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm", *IEEE Transactions on Systems Man and Cybernetics - Part A*, Vol. 38, No. 1, pp. 1-20, January 2008.
12. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002.
13. D.W. Corne, J.D. Knowles, and M.J. Oates, "The pareto-envelope based selection algorithm for multiobjective optimisation", in M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H-P. Schwefel, (eds.) *Parallel Problem Solving from Nature - PPSN VI*, Springer Lecture Notes in Computer Science, pp. 869–878, 2000.
14. X. Xie and G. Beni, "Validity measure for fuzzy clustering", *IEEE Trans. Pattern Anal. Machine Learning*, Vol. 3, pp. 841–846, (1991).
15. J. C. Bezdek, "Cluster validity with fuzzy sets", *Journal of Cybernetics*, (3) 58–72, (1974).
16. S. Chu *et al.* "The transcriptional program of sporulation in budding yeast", *Science*, 282, 699–705, 1998.
17. Y. Sawaragi, H. Nakayama, and T. Tanino, "Theory of multiobjective optimization" (vol. 176 of *Mathematics in Science and Engineering*). Orlando, FL: Academic Press Inc., 1985.
18. K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001.
19. C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, 2007.
20. R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, 11(4) (1997) 341–359.
21. R. Storn, K. V. Price, and J. Lampinen, *Differential Evolution - A Practical Approach to Global Optimization*, Springer, Berlin, 2005.
22. C. A. Mattson, A. A. Mullur, and A. Messac, "Smart Pareto filter: Obtaining a minimal representation of multiobjective design space," *Eng. Optim.*, vol. 36, no. 6, pp. 721–740, 2004.
23. J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Proc. 8th Int. Conf. Parallel Problem Solving From Nature*, pp. 722–731, 2004.
24. R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic," *J. Royal Statist. Soc.: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.
25. L. Hubert and P. Arabie, "Comparing partitions", *Journal of Classification*, 193–218, 1985.
26. W. M. Rand, "Objective criteria for the evaluation of clustering methods", *Journal of the American Statistical Association*, 66, 846–850, 1971.

27. P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, 1987.

28. C. Blake, E. Keough and C.J. Merz, UCI repository of machine learning database (1998). <http://www.ics.uci.edu/~mllearn/MLrepository.html>

29. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Second Edition, Elsevier Academic Press, 2003.

30. D. A. Keim and H.-P. Kriegel, "Visualization techniques for mining large databases: a comparison", *IEEE Transactions on*

*Knowledge and Data Engineering*, v.8 n.6, p.923-938, December 1996.

31. <http://cmgm.stanford.edu/pbrown/sporulation>

32. <http://dbkgroup.org/handl/mock/>

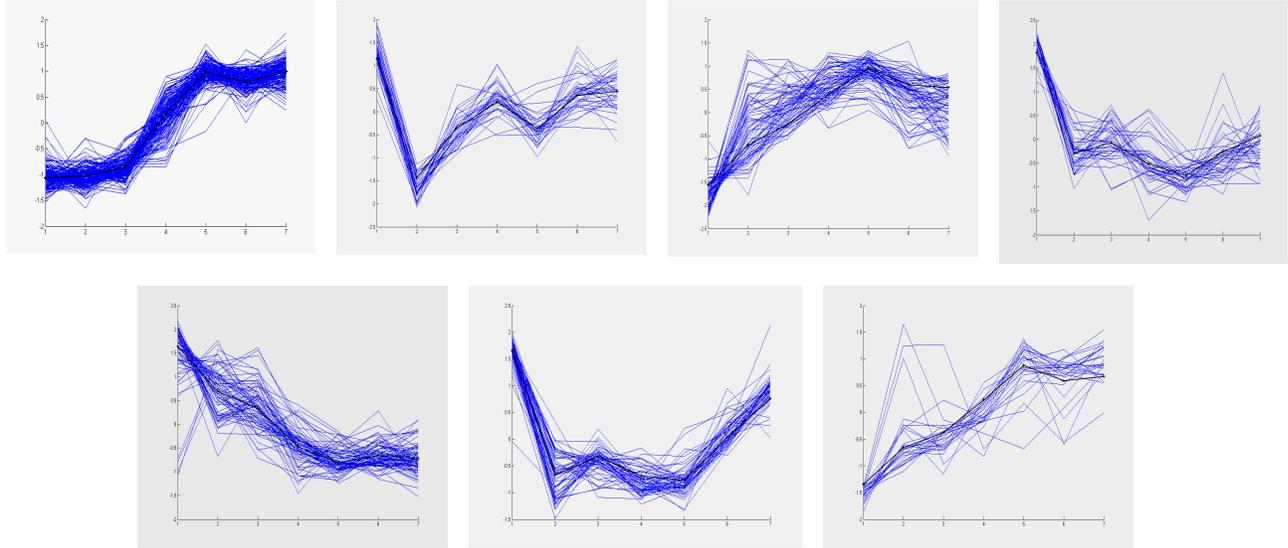


FIGURE 1. PARALLELCOORDS PLOTS OF CLUSTER 1 TO 4 (FROM LEFT TO RIGHT ON TOP), AND FOR CLUSTERS 5, 6, AND 7 BELOW.

TABLE 2. AVERAGE SILHOUETTE INDEX AND NUMBER OF CLUSTERS FOUND AND STANDARD DEVIATIONS (IN PARENTHESES) BY SIX ALGORITHMS OVER 30 INDEPENDENT RUNS ON TEN DATASETS.

Dataset	Algorithms											
	MODE		DEMO		PDE		NSDE		NSGA2		MOCK	
	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index
Dataset_1	<b>9.12</b> (1.46)	<b>0.670726</b> (0.0321)	9.43 (0.843)	0.670523 (0.08263)	9.83 (0.35)	0.656594 (0.05372)	8.81 (2.837)	0.661661 (0.00483)	9.37 (1.72)	0.669317 (0.0892)	8.52 (2.81)	0.66342 (0.0736)
Dataset_2	3.36 (0.65)	0.678734 (0.03716)	3.74 (0.363)	0.670530 (0.05381)	3.15 (0.726)	0.663812 (0.00366)	<b>3.07</b> (0.052)	<b>0.683382</b> (0.00082)	3.16 (0.072)	0.674393 (0.00927)	3.33 (1.03)	0.658921 (0.004731)
Dataset_3	4.14 (0.36)	0.771363 (0.00938)	<b>4.09</b> (0.24)	<b>0.774611</b> (0.002376)	4.38 (1.09)	0.766945 (0.04824)	4.17 (0.73)	0.767333 (0.004712)	3.57 (0.51)	0.765691 (0.005686)	3.78 (1.25)	0.768419 (0.006721)
Dataset_4	<b>6.04</b> (0.25)	<b>0.849184</b> (0.00472)	6.13 (1.27)	0.838303 (0.056741)	6.15 (0.79)	0.824910 (0.00721)	6.31 (1.46)	0.840613 (0.04752)	6.28 (0.46)	0.827618 (0.02871)	6.08 (0.51)	0.832527 (0.007825)
Dataset_5	9.24 (3.89)	0.778102 (0.04363)	<b>10.03</b> (0.37)	<b>0.778736</b> (0.008125)	9.26 (2.88)	0.746241 (0.00673)	11.45 (1.68)	0.777406 (0.05345)	12.43 (0.939)	0.768379 (0.005384)	10.41 (0.80)	0.769342 (0.006208)
Dataset_6	5.19 (0.93)	0.643316 (0.00381)	5.62 (0.867)	0.637986 (0.008112)	<b>5.08</b> (0.35)	<b>0.645446</b> (0.05524)	5.26 (0.78)	0.640832 (0.004983)	4.65 (1.58)	0.642091 (0.002833)	5.16 (0.38)	0.640957 (0.008349)
Iris	2.31 (0.76)	0.606353 (0.03483)	<b>2.97</b> (0.40)	<b>0.606864</b> (0.03234)	2.25 (0.324)	0.569675 (0.04752)	2.14 (0.58)	0.600304 (0.004618)	2.16 (1.06)	0.566613 (0.082651)	3.05 (0.37)	0.6003725 (0.005129)
Wine	<b>3.16</b> (0.46)	<b>0.582197</b> (0.00427)	3.65 (0.83)	0.568391 (0.007473)	4.01 (1.35)	0.525840 (0.01213)	3.52 (1.20)	0.5457383 (0.009497)	3.88 (0.67)	0.5767342 (0.009415)	3.59 (0.46)	0.576834 (0.000812)
Breast Cancer	<b>2.08</b> (0.38)	<b>0.648297</b> (0.00734)	2.68 (0.64)	0.609123 (0.57813)	2.15 (0.46)	0.619219 (0.00567)	2.33 (0.67)	0.628352 (0.006782)	2.57 (0.60)	0.6004642 (0.004561)	2.10 (0.53)	0.626719 (0.01094)
Yeast Sporulation	<b>7.08</b> (0.12)	<b>0.676434</b> (0.00072)	6.34 (0.32)	0.558619 (0.057832)	8.36 (2.81)	0.595367 (0.00721)	7.24 (1.04)	0.604513 (0.005728)	7.22 (0.68)	0.641306 (0.04813)	6.67 (0.857)	0.613567 (0.005738)

**TABLE 3.** MEAN VALUE OF ADJUSTED RAND INDEX FOUND AND STANDARD DEVIATIONS (IN PARENTHESES) BY DIFFERENT ALGORITHMS OVER 30 INDEPENDENT RUNS ON NINE DATASETS.

Dataset	Algorithms					
	MODE	DEMO	PDE	NSDE	NSGA2	MOCK
Dataset_1	<b>0.846199</b> (0.031257)	0.828437 (0.046182)	0.719584 (0.00563741)	0.819794 (0.035285)	0.802180 (0.004782)	0.810934 (0.0059348)
Dataset_2	0.847621 (0.006312)	0.9273464 (0.0008573)	0.9372649 (0.036451)	<b>1.000000</b>	0.9378123 (0.006821)	0.946547 (0.004536)
Dataset_3	0.951786 (0.004827)	<b>1.000000</b>	0.9758732 (0.05736)	0.894635 (0.005736)	0.963841 (0.0046719)	0.878732 (0.0712523)
Dataset_4	<b>1.000000</b>	0.857463 (0.065639)	0.840953 (0.076829)	0.919843 (0.0121436)	0.957818 (0.004678)	0.978761 (0.006734)
Dataset_5	0.983785 (0.076764)	<b>0.993173</b> (0.089371)	0.876710 (0.023376)	0.982013 (0.084372)	0.947641 (0.006646)	0.9454568 (0.0012043)
Dataset_6	0.881413 (0.05983)	0.881136 (0.078348)	<b>0.884930</b> (0.007846)	0.880265 (0.056347)	0.881395 (0.056483)	0.910294 (0.016743)
Iris	0.738626 (0.0756779)	<b>0.748784</b> (0.067457)	0.709036 (0.025739)	0.738960 (0.001436)	0.715898 (0.005739)	0.786574 (0.075763)
Wine	<b>0.875849</b> (0.0087642)	0.858876 (0.0035287)	0.8265764 (0.0032429)	0.845365 (0.0065761)	0.828645 (0.0074653)	0.864764 (0.0034398)
Breast Cancer	0.956456 (0.0056453)	0.912173 (0.0043247)	0.937857 (0.0087743)	<b>0.950965</b> (0.0065682)	0.944236 (0.006521)	0.9465731 (0.006748)

**TABLE 4.** UNPAIRED T-TEST RESULTS FOR SILHOUETTE INDEX

Dataset	Std. Err	<i>t</i>	95% Conf. Intvl	Two-tailed <i>P</i>	Significance
Dataset_1	0.001	7.1968	-0.0121 to -0.0068	< 0.0001	Extremely Significant
Dataset_2	0.002	3.8990	-0.0129 to -0.0040	< 0.0001	Extremely Significant
Dataset_3	0.007	34.9267	-0.2665 to -0.2373	< 0.0001	Extremely Significant
Dataset_4	0.001	3.0961	-0.0051 to -0.0010	0.0037	Very Significant
Dataset_5	0.003	3.0684	-0.0156 to -0.0032	0.0040	Very Significant
Dataset_6	0.002	3.0584	-0.0109 to -0.0022	0.0041	Very Significant
Iris	0.009	1.3744	-0.0309 to 0.0059	0.1774	Not Significant
Wine	0.003	2.3999	-0.0118 to -0.0010	0.0214	Significant
Breast Cancer	0.009	1.3744	-0.0309 to 0.0059	0.1774	Not Significant
Yeast Sporulation	0.003	2.3999	-0.0118 to -0.0010	0.0214	Significant