Data Clustering Using Multi-objective Differential Evolution Algorithms

Kaushik Suresh, Debarati Kundu, Sayan Ghosh, Swagatam Das

Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata, India.

Ajith Abraham*

†*School of Computer Science,Dalian Maritime University, 116024 Dalian, China.*
‡*Machine Intelligence Research Labs (MIR Labs),*
Scientific Network for Innovation and Research Excellence, USA.
http://www.mirlabs.org, ajith.abraham@ieee.org

Abstract. The article considers the task of fuzzy clustering in a multi-objective optimization (MO) framework. It compares the relative performance of four recently developed multi-objective variants of Differential Evolution (DE) on 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 for the candidates is used for DE. A comparative study of four DE variants with two most well-known MO clustering techniques, namely the NSGA II (Non Dominated Sorting GA) and MOCK (Multi-Objective Clustering with an unknown number of clusters K) is also undertaken. Experimental results reported for six artificial and four real life datasets (including a microarray dataset of budding yeast) of varying range of complexities indicates that DE can serve as a promising algorithm for devising MO clustering techniques.

Keywords: Differential Evolution, Multi-objective optimization, Fuzzy clustering, Micro-array data clustering

Address for correspondence: Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, USA.

*Supported by National Science Foundation of China Grant 60873054

1. Introduction

Optimization-based clustering algorithms greatly rely on a cluster validity function (optimization criterion) the optimum of which appears as a proxy for the unknown "correct classification" in a previously unhandled dataset. Different formulations of the clustering problem vary with regard to the optimization criterion used. Most existing clustering methods, however, attempt, explicitly or otherwise, to optimize just one such clustering criterion modeled by a single cluster validity index. This often results in considerable discrepancies observable between the solutions offered 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 for 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 it would not recognize them as good and would almost surely discard them.

Although there has been a plethora of papers reporting several single-objective evolutionary clustering techniques, very few research works have so far been undertaken towards the application of evolutionary multi-objective optimization algorithms (EMOA) for pattern clustering. Two most prominent multi-objective clustering techniques are based on NSGA II [1] and PESA II [2].

A state-of-the-art literature survey indicates that DE [3] has already proved itself as a promising candidate in the field of Evolutionary Multi-objective Optimization (EMO) [4 - 6]. Unlike NSGA II or PESA II, 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 aims at comparing 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) [7], the Multi-objective DE (MODE) [8], DE for Multi-objective Optimization (DEMO) [9], and Non-Dominated Sorting DE (NSDE) [10]. 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 any further on any objective without degrading it on another. For assessing the clustering performance of the DE-variants, we consider the Xie-Beni index [11] and the Fuzzy C Means (FCM) measure ( ) [12] as the objective functions. Note that any other and any number of objective functions could be used in the proposed MOO clustering framework. The best clustering solutions have been selected from the Pareto-optimal set using the crowding distance [13] method for all DE-variants. The performance of the multi-objective DE-variants has also been contrasted with two best-known EMOA-based clustering methods till date. The first one of these is MOCK by Handl and Knowles [14] 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 [15]. 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].

2. Multi-objective Optimization with DE

In this section we outline four most frequently used variants of DE, specially devised for tackling the MOO problems.

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. Consider for example, the design of an automobile. Possible objectives could be: minimize cost, maximize speed, minimize fuel consumption and maximize luxury. These goals are clearly conflicting and, therefore, there is no single optimum to be found. The field of Multi-objective Optimization (MO) deals with such simultaneous optimization of multiple and possibly competing objective functions. The MO problems tend to be characterized by a family of alternatives that 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:

Definition 2.1. Consider, without loss of generality, the following multi-objective optimization problem with m decision variables (parameters) and n objectives y:

$$Maximize\vec{Y} = f(\vec{Y}) = (f_1(x_1, \cdots, x_m), \cdots, f_n(x_1, \cdots, x_m)) \tag{1}$$

$\vec{X} = [x_1, \cdots, x_m]^T \in P$ and $\vec{Y} = [y_1, \cdots, y_m]^T \in O$ 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:

$$\forall i \in \{1, \cdots, n\} \quad f_i(\vec{A}) \succeq f_i(\vec{B}) \wedge \exists j \in \{1, \cdots, n\} \quad f_j(\vec{A}) > f_j(\vec{B}) \tag{2}$$

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

Definition 2.2. 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,

$$\nexists \vec{A'} \in P' : P' \succ P \tag{3}$$

(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 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) t is a D-dimensional vector containing a set of D optimization parameters:

$$\vec{Z}_i(t) = [Z_{i,1}(t), Z_{i,2}(t), \cdots, Z_{i,D}(t)] \tag{4}$$

Now in each generation (or one iteration of the algorithm) to change the population members $\vec{Z}_i(t)$ (say), a donor vector $\vec{Y}_i(t)$ is created. It is the method of creating this donor vector that distinguishes the various DE schemes. In one of the earliest variants of DE, now called DE/rand/1 scheme, to create $\vec{Y}_i(t)$ for each i-th member, three other parameter vectors (say the r_1, r_2, and r_3-th vectors such that r_1, r_2 and 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 $\vec{Y}_i(t)$. The process for the j-th component of the i-th vector may be expressed as,

$$\vec{Y}_{i,j}(t) = Z_{r_1,j}(t) + F(Z_{r_2,j}(t) - Z_{r_3,j}(t)) \tag{5}$$

Next a crossover operation takes place to increase the potential diversity of the population. The DE-family mostly uses two kinds of crossover schemes, namely ‘exponential’ and ‘binomial’ [3]. To save space, we here briefly describe the binomial crossover, which is also employed by the modified DE algorithm. The binomial crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the Cr value. In this case the number of parameters inherited from the mutant has a (nearly) binomial distribution. Thus for each target vector $\vec{Z}_i(t)$, a trial vector $\vec{R}_i(t)$ is created in the following fashion:

$$R_i(t) = \begin{cases} Y_{i,j}(t) & \text{if } rand_j(0,1) \geq Cr \text{ or } j = rn(i) \\ Z_{i,j}(t) & \text{if } rand_j(0,1) > Cr \text{ or } j \neq rn(i) \end{cases} \tag{6}$$

for $j = 1, 2, .., D$ and $rand_j(0,1) \in [0,1]$ is the j-th evaluation of a uniform random number generator. $rn(i) \in [1, 2, \cdots, D]$ is a randomly chosen index which ensures that $\vec{R}_i(t)$ gets at least one component from $\vec{Z}_i(t)$. To keep the population size constant over subsequent generations, the 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:

$$\vec{Z}_i(t+1) = \begin{cases} \vec{R}_i(t) & \text{if } \vec{R}_i(t) \geq \vec{Z}_i(t) \\ \vec{Z}_i(t) & \text{if } \vec{R}_i(t) > \vec{Z}_i(t) \end{cases} \tag{7}$$

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

2.3. The Multi-objective Variants of DE

2.3.1. Pareto Differential Evolution Algorithm (PDE)

One of the recent approaches of evolutionary multi-objective optimization is Pareto differential evolution algorithm [7]. This algorithm was designed for EMO problems with continuous variables and achieved very competitive results compared to other evolution algorithms in EMO literature. However, there is no obvious way to select best crossover and mutation rates apart from running the algorithm with different rates. It handles only one (main) population. Reproduction is undertaken only among non-dominated solutions, and offspring are placed in the population if they dominate the main parent. A distance relationship is used to maintain diversity.

The PDE algorithm is similar to the DE algorithm with some modifications:

- The initial population is initialized according to a Gaussian distribution $N(0.5, 0.15)$.
- The step-length parameter F is generated from a Gaussian distribution $N(0, 1)$.
- Reproduction is undertaken only among non-dominated solutions in each generation.
- The boundary constraints are preserved either by reversing the sign if the variable is less than 0 or keeping subtracting 1 if it is greater than 1 until the variable is within its boundaries.
- Offspring are placed into the population if they dominate the main parent.

In PDE, the 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.

2.3.2. The Multi-Objective Differential Evolution (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. Xue et al. used MODE to solve five high dimensionality unconstrained problems with 250,000 evaluations and the results are compared only to those obtained by SPEA [19].

2.3.3. Differential Evolution for Multi-objective Optimization (DEMO)

DEMO [9] 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 [1]) of the extended population promotes the uniform spread of solutions.

DEMO is implemented in three variants (DEMO/parent, DEMO/closest/dec and DEMO/closest/obj) [9]. Below we provide a pseudo-code for MODE/parent:

1. Evaluate the initial population P of random individuals.

```
2. While stopping criterion not met, do:
    2.1 For each individual $\vec{X}_i$ $(i = 1, \cdots, NP)$ from $P$ repeat:
        (a) Create candidate $\vec{U}_i$ from parent $\vec{X}_i$.
        (b) Evaluate the candidate.
        (c) If the candidate dominates the parent, the candidate replaces the parent.
            If the parent dominates the candidate, the candidate is discarded.
            Otherwise, the candidate is added in the population.
    2.2. If the population has more than population size $NP$ individuals, truncate it.
    2.3. Randomly enumerate the individuals in $P$.
```

DEMO is compared to NSGA-II [1], PDEA [20], PAES [2], SPEA [19], and MODE [8] over five high-dimensionality unconstrained problems and was found to meet or beat the competitors in majority of the cases.

2.3.4. Non-dominated Sorting Differential Evolution (NSDE)

This approach was proposed in [10]. It is a simple modification of the NSGA-II [1]. The only difference between this approach and the NSGA-II is in the method for generating new individuals. The NSGA-II uses a real-coded crossover and mutation operator, but in the NSDE, these operators were replaced with the operators of Differential Evolution. In the NSGA-II the better individuals remain unknown until all candidates are sorted and assigned a crowding distance and non-domination level. Parents are selected for mating using a tournament selection operator which uses the rank and crowding distance of individuals. Within the NSGA-II framework, the DE variants are used to generate N offspring from the selected parents. The offspring 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. The results of the NSDE outperformed those produced by the NSGA-II.

3. Multi-objective Clustering Scheme

In this section we step-wise describe a multi-objective framework for clustering various datasets.

3.1. Search-variable Representation

To search for the globally optimal solution to a problem using DE, parameters of the problem have to be represented by real numbers. In the proposed method, if there are n document vectors, each D-dimensional, and if the user-specified maximum number of clusters is k then each chromosome is a vector of real numbers of dimension $k \times D$. The entries are reserved for k number of p-dimensional cluster centers. A single chromosome can be shown as:

$$\vec{X}_i(t) = \boxed{\vec{m}_1}\boxed{\vec{m}_2}\cdots\cdots\boxed{\vec{m}_k}$$

The chromosome encoding scheme may be illustrated with the following example:

Example 1: Let $N = 2$ and $k = 3$, i.e., the space is two dimensional and the number of clusters being considered is three. Then the chromosome $\vec{Z}_i(t) = [61.6, 75.3, 19.3, 10.7, 18.3, 30.2]$ represents the three cluster centers (61.6, 75.3), (19.3, 10.7), and (18.3, 30.2).

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. The selection of objectives should be such that they can balance each other critically and are contradictory in nature. 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. For convenience we present the two functions here once again. The FCM measure J_q may be defined as:

$$J_q = \sum_{j=1}^{n} \sum_{i=1}^{k} u_{ij}^{q} \cdot d^2(\vec{Z}_j, \vec{m}_i), 1 \leq q \leq \infty \tag{8}$$

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. Please note that in this work we mostly use the Euclidean distance and the data have been normalized, so that the high-valued features of broad range may not suppress the differences between low magnitude features.

The XB index is defined as a function of the ratio of the total variation σ (which in fact is similar to J_q) to the minimum separation sep of the clusters. Here σ and sep may be written as:

$$\sigma = \sum_{i=1}^{k} \sum_{p=1}^{n} u_{ip}^{2} \cdot d(\vec{m}_i, \vec{Z}_j) \tag{9}$$

and

$$sep(Z) = \min_{i \neq j} \left\{ d^2(\vec{m}_i, \vec{m}_j) \right\} \tag{10}$$

The XB index is then written as:

$$XB_q = \frac{\sigma}{n \times sep(Z)} = \frac{\sum_{i=1}^{k} \sum_{p=1}^{n} u_{ip}^{q} \cdot d^2(\vec{m}_i, \vec{Z}_p)}{\min_{i \neq j} \left\{ d^2(\vec{Z}_i, \vec{Z}_j) \right\}} \tag{11}$$

Note that when the partitioning is compact and the individual clusters are well separated, value of σ 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 (8) and (11), the centers encoded in a DE vector are first extracted. Let the set of centers be denoted by $\{\vec{m}_1, \vec{m}_2, \cdots, \vec{m}_k\}$. The membership value of the j-th pattern in i-th cluster u_{ij}, $i = 1, 2, \cdots, k$ and $j = 1, 2, \cdots, n$ are computed as:

$$u_i j = \frac{1}{\sum_{p=1}^{k} \left(\frac{d(\vec{m}_i, \vec{Z}_j)}{d(\vec{m}_p, \vec{Z}_j)} \right)^{\frac{2}{q-1}}} \tag{12}$$

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

$$\vec{m}_p = \frac{\sum_{j=1}^{n} (u_{pj})^q \cdot \vec{(Z)}_j}{\sum_{j=1}^{n} (u_{pj})^q} \tag{13}$$

and the membership values are recomputed.

Note that the performance of multi-objective clustering highly depends on the choice of objectives which should have as much conflict among themselves as possible. In this article the XB and J_q (FCM) indices have been chosen as the two objectives to be optimized. Equation (8) reveals that J_q calculates the global cluster variance, i.e., it considers the within cluster variance summed up over all the clusters. Lower value of J_q implies better clustering solution. On the other hand, the XB index (eqn. (11)) is a combination of global (numerator) and local (denominator) situations. Although the numerator of XB index is similar to J_q, the denominator contains an additional term (sep) representing the separation between the two nearest clusters. Therefore, XB index is minimized by minimizing σ (or J_q), and by maximizing sep. These two terms may not attain their best values for the same partitioning when the data has complex and overlapping clusters, such as the micro-array data and many other real life datasets, considering J_q and XB (or in effect σ and sep) will provide a set of alternate clustering solutions for the data.

Figure 1 shows, just for the sake of illustration, the final Pareto-optimal front (composed of non-dominated solutions) of one of the runs of the MODE algorithm for the artificial dataset_3 (described in the next section), to demonstrate the contradictory nature of J_q and XB indices.

3.3. Selecting the Best Solutions 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 [21-24]. These works have primarily dealt with the reduction of the size of the approximation set in absence of additional expert's knowledge. For choosing the most interesting solutions from the Pareto front, we apply Tibshirani et al.'s Gap statistic [25], a statistical method to determine the number of clusters in a dataset. 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.

3.4. Evaluating the Clustering Quality

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

indicates a measure of the confidence of belongingness, and it is defined as:

$$s(\vec{Z}_j) = \frac{b(\vec{Z}_j) - a(\vec{Z}_j)}{\max(a(\vec{Z}_j), b(\vec{Z}_j))} \tag{14}$$

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, $r = 1, 2, \cdots, k$ 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) indicates that the data point $s(\vec{Z}_j)$ is well clustered. Value of $s(\vec{Z}_j)$ around 0 means that the data point lies 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 silhouete index $s(C)$ of a clustering solution $C = \{C_1, C_2, \cdots, C_k\}$ is defined as the mean silhouette width over all the data points:

$$s(C) = \frac{1}{n} \sum_{j=1}^{n} s(\vec{Z}_j) \tag{15}$$

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.

The adjusted Rand index comes as a generalization of the Rand Index [28]. It introduces a statistically induced normalization in order to yield values close to 0 for random partitions. Using a representation based on contingency tables, the Adjusted Rand Index is given by:

$$R = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \left[\sum_i \binom{n_{i+}}{2} \cdot \sum_j \binom{n_{j+}}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{n_{i+}}{2} + \sum_j \binom{n_{j+}}{2}\right] - \left[\sum_i \binom{n_{i+}}{2} \cdot \sum_j \binom{n_{j+}}{2}\right] / \binom{n}{2}} \tag{16}$$

where n is the total number of data points, and n_{ij} is the number of data points classified into class i in the experimental classification and into class j in the real classification. Also $n_{i+} = \sum_j n_{ij}$ is the number of objects classified into cluster i in the experiment, and $n_{j+} = \sum_j n_{ij}$ is the number of objects classified into class j in the actually known classification.

4. Experimental Results

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, where the first five datasets are in two dimensions and the last one is three dimensional 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, glass and the yeast sporulation data. Except for the yeast sporulation, the other four real-life data have been taken from the publicly available machine learning data repository [34].

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

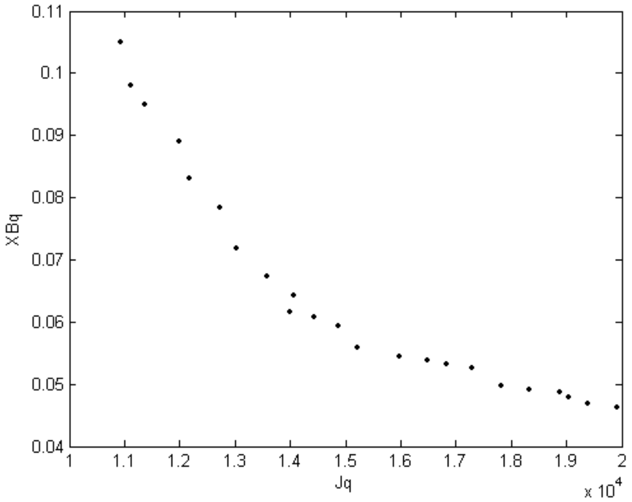


Figure 1. Non-dominated Pareto front for artificial dataset_3.

publicly from the website [35]. 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. After conducting a series of experiments we find that choosing the value of scale factor F randomly between 0.5 and 1 while keeping the value of Cr at 0.9 gives reasonably good results for all the four DE variants considered here. In order to make the comparison fair enough, we ensure that in each run over each problem, all the multi-objective DE variants may start from the same initial population. In this way, any performance difference between these algorithms is actually attributed to the internal operators of these algorithms and not to the initial population which may accidentally hit a very good solution for some algorithm.

The 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 = $\frac{1}{Chromosome\_length}$. Please note that the four DE variants and the NSGA II use the same parameter representation scheme. Clustering with MOCK was performed with the source codes available from [36].

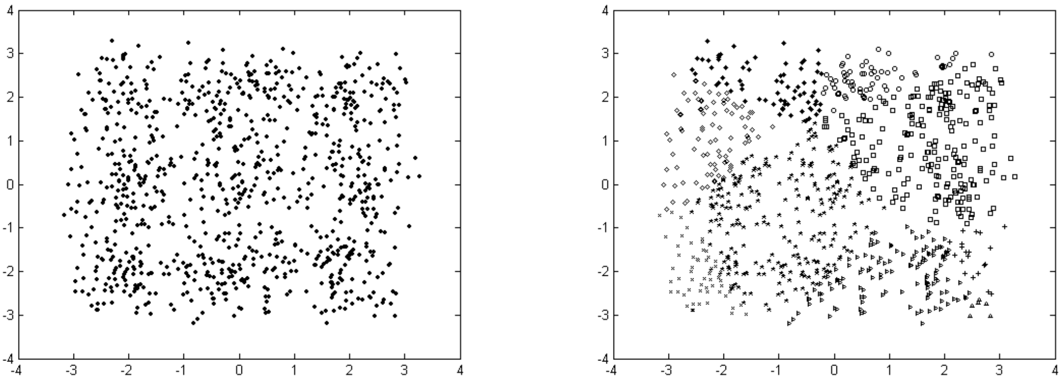
Table 1. Details of the datasets used.

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
Glass	214	6	9
Yeast Sporulation	474	7	7

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 table also shows the mean number of classes determined by each algorithm. All the results presented in this table are averages over 30 independent runs of each algorithm. The best entries have been marked in boldface in each row. Table 3 is identical in spirit to Table 2, only difference being that this table enlists the adjusted rand index values. Missing values of standard deviations in this table indicate zero standard deviations. The Yeast sporulation data is excluded from this table as no standard nominal classification is known for this dataset. The effectiveness of the clustering on this data has been established by finding the biological interpretation of the clusters by using Gene Ontology (GO) [30] terms. Please see section 4.4 for details of this issue. Tables 3 and 4 show 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 both average Silhouette index and adjusted rand index. For all cases in Tables 3 and 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 and adjusted rand index in a statistically significant fashion. The six unlabelled artificial datasets and the corresponding clustered data with the best performing algorithm (which happens to be one of the four multi-objective DE variants) have been depicted in Figures 2 to 7.

Note that the three clusters identified in iris data corresponds to three species of Iris flowers: Iris setosa, Iris virginica and Iris versicolor. One of the clusters correspond to the Setosa class is linearly separable while the other two clusters corresponding to the classes Veriscolor and Virginica are considerably overlapping. Due to the overlapping of the two clusters, many of the evolutional learning processes implicitly assume the number of naturally occurring groups in this dataset to be 2. This results into a relatively lower value of the adjusted rand index, which compares the experimental clustering with the nominal clustering with the knowledge of human experts. The fact is visible from the 8-th column of



(a) The unlabelled dataset_1

(b) Clustering with MODE

Figure 2. Clustering result for artificial dataset_1.

Table 3 as well. DEMO yielded the best results for this data set as judged by the silhouette and adjusted rand indices.

The wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three clusters of wines. In a classification context, this is a well posed problem with "well behaved" cluster structures. The Wisconsin breast cancer database contains 9 relevant features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. The dataset has two classes. The objective is to classify each data vector into benign (239 objects) or malignant tumors (444 objects).Nearly all algorithms yielded good results on this dataset with MODE showing the best clustering performance with respect to both silhouette index and adjusted rand index.

The glass data were sampled from six different type of glass: building windows float processed (70 objects), building windows non float processed (76 objects), vehicle windows float processed (17 objects), containers (13 objects), tableware (9 objects), headlamps (29 objects) with nine features each. These are refractive index, Sodium, Magnesium, Aluminum, Silicon, Potassium, Calcium, Barium and Iron. MODE clustered the dataset most accurately as judged by the two validity indices used here.

A detailed interpretation of clustering of the microarray dataset (yeast sporulation) has been provided in the following section.

4.4. Significance and Validation of Microarray Data Clustering Results

In this section, 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[31]) and the heatmap [32] plot in MATLAB 7.0.4 version. Parallel coordinates is a common way of visualizing high-dimensional geometry and analyzing multivariate data. To show a set of points in an n-dimensional space, a backdrop

Table 2. Details of the datasets used.

Dataset	Silhouette Index					
	MODE	DEMO	PDE	NSDE	NSGA2	MOCK
Dataset_1	0.670726 (0.0321)	0.670523 (0.08263)	0.656594 (0.05372)	0.661661 (0.00483)	0.669317 (0.0892)	0.66342 (0.0736)
Dataset_2	0.678734 (0.03716)	0.670530 (0.05381)	0.663812 (0.00366)	0.683382 (0.00082)	0.674393 (0.00927)	0.658921 (0.004731)
Dataset_3	0.771363 (0.00938)	0.774611 (0.002376)	0.766945 (0.04824)	0.767333 (0.004712)	0.765691 (0.005686)	0.768419 (0.006721)
Dataset_4	0.849184 (0.00472)	0.838303 (0.056741)	0.824910 (0.00721)	0.840613 (0.04752)	0.827618 (0.02871)	0.832527 (0.007825)
Dataset_5	0.778102 (0.04363)	0.778736 (0.008125)	0.746241 (0.00673)	0.777406 (0.05345)	0.768379 (0.005384)	0.769342 (0.006208)
Dataset_6	0.643316 (0.00381)	0.637986 (0.008112)	0.645446 (0.05524)	0.640832 (0.004983)	0.642091 (0.002833)	0.640957 (0.008349)
Iris	0.606353 (0.03483)	0.606864 (0.03234)	0.569675 (0.04752)	0.600304 (0.004618)	0.566613 (0.082651)	0.6003725 (0.005129)
Wine	0.582197 (0.00427)	0.568391 (0.007473)	0.525840 (0.01213)	0.5457383 (0.009497)	0.5767342 (0.009415)	0.576834 (0.000812)
Breast Cancer	0.648297 (0.00734)	0.609123 (0.57813)	0.619219 (0.00567)	0.628352 (0.006782)	0.6004642 (0.004561)	0.626719 (0.01094)
Glass	0.743521 (0.01142)	0.738639 (0.011253)	0.703623 (0.04416)	0.739832 (0.011174)	0.682434 (0.008325)	0.740636 (0.023733)
Yeast Sporulation	0.676434 (0.00072)	0.558619 (0.057832)	0.595367 (0.00721)	0.604513 (0.005728)	0.641306 (0.04813)	0.613567 (0.005738)

Table 3. Mean value of adjusted rand index found and standard deviations (in parentheses) by six contestant algorithms over 30 independent runs on ten datasets.

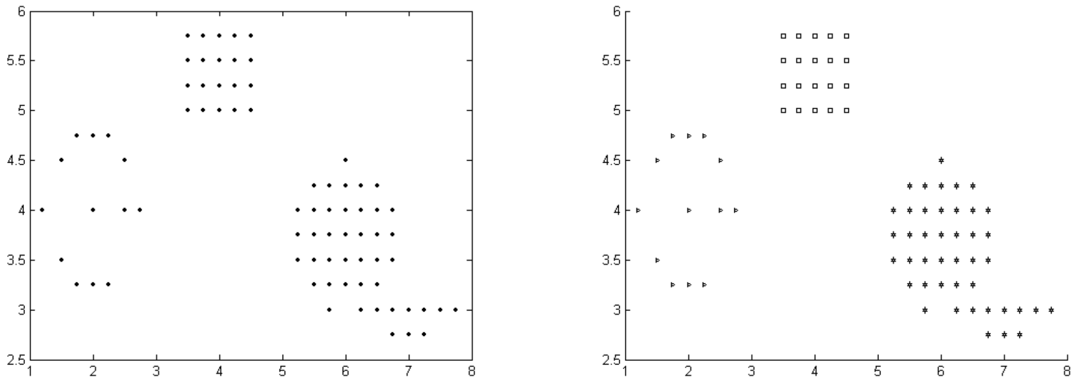
Dataset	Algorithms					
	MODE	DEMO	PDE	NSDE	NSGA2	MOCK
Dataset_1	0.846199	0.828437	0.719584	0.819794	0.802180	0.810934
	(0.031257)	(0.046182)	(0.00563741)	(0.035285)	(0.004782)	(0.0059348)
Dataset_2	0.847621	0.9273464	0.9372649	1.000000	0.9378123	0.946547
	(0.006312)	(0.0008573)	(0.036451)		(0.006821)	(0.004536)
Dataset_3	0.951786	1.000000	0.9758732	0.894635	0.963841	0.878732
	(0.004827)		(0.05736)	(0.005736)	(0.0046719)	(0.0712523)
Dataset_4	1.000000	0.857463	0.840953	0.919843	0.957818	0.978761
		(0.065639)	(0.076829)	(0.0121436)	(0.004678)	(0.006734)
Dataset_5	0.983785	0.993173	0.876710	0.982013	0.947641	0.9454568
	(0.076764)	(0.089371)	(0.023376)	(0.084372)	(0.006646)	(0.0012043)
Dataset_6	0.881413	0.881136	0.884930	0.880265	0.881395	0.910294
	(0.05983)	(0.078348)	(0.007846)	(0.056347)	(0.056483)	(0.016743)
Iris	0.738626	0.748784	0.709036	0.738960	0.715898	0.786574
	(0.0756779)	(0.067457)	(0.025739)	(0.001436)	(0.005739)	(0.075763)
Wine	0.875849	0.858876	0.8265764	0.845365	0.828645	0.864764
	(0.0087642)	(0.0035287)	(0.0032429)	(0.0065761)	(0.0074653)	(0.0034398)
Breast	0.956456	0.912173	0.937857	0.950965	0.944236	0.9465731
Cancer	(0.0056453)	(0.0043247)	(0.0087743)	(0.0065682)	(0.006521)	(0.006748)
Glass	0.8412155	0.835356	0.823165	0.838960	0.825398	0.836574
	(0.00694374)	(0.0709425)	(0.0523547)	(0.0263573)	(0.0142463)	(0.0074782)

Table 4. Unpaired t-test results for silhouette index.

Dataset	Std. Err	t	95% Conf. Intvl	Two-tailed P	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
Glass	0.000	5.9997	0.001922 to 0.003847	< 0.0001	Extremely Significant
Yeast Sporulation	0.003	2.3999	-0.0118 to -0.0010	0.0214	Significant

Table 5. Unpaired t-test results for adjusted rand index.

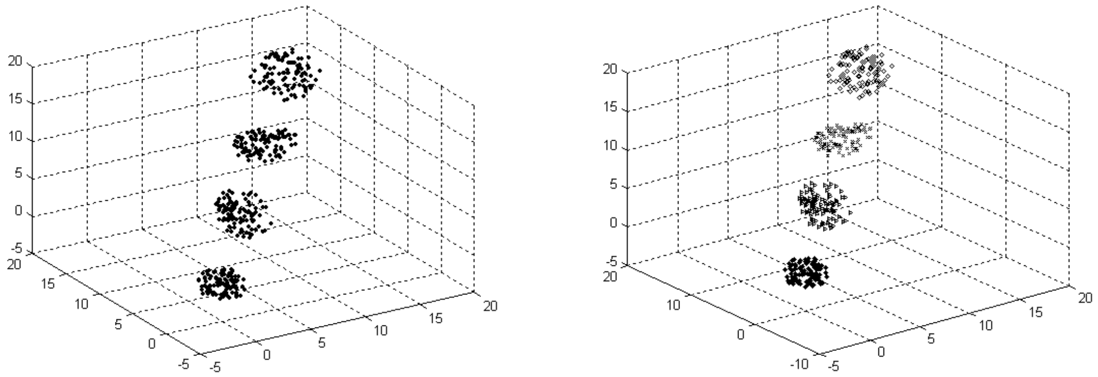
Dataset	Std. Err	t	95% Conf. Intvl	Two-tailed P	Significance
Dataset_1	0.021	2.9201	-0.1050 to -0.0189	0.0059	Very significant
Dataset_2	0.013	5.0453	-0.0922 to -0.0394	< 0.0001	Extremely Significant
Dataset_3	0.002	17.965	-0.0452 to -0.0360	< 0.0001	Extremely Significant
Dataset_4	0.005	6.4431	-0.0419 to -0.0219	< 0.0001	Extremely Significant
Dataset_5	0.009	1.3744	-0.0309 to 0.0059	0.1774	Not Significant
Dataset_6	0.003	2.3999	-0.0118 to -0.0010	0.0214	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
Glass	0.002	2.4912	0.0009120 to 0.0083709	0.0156	Significant



(a) The unlabelled dataset_2

(b) Clustering with NSDE

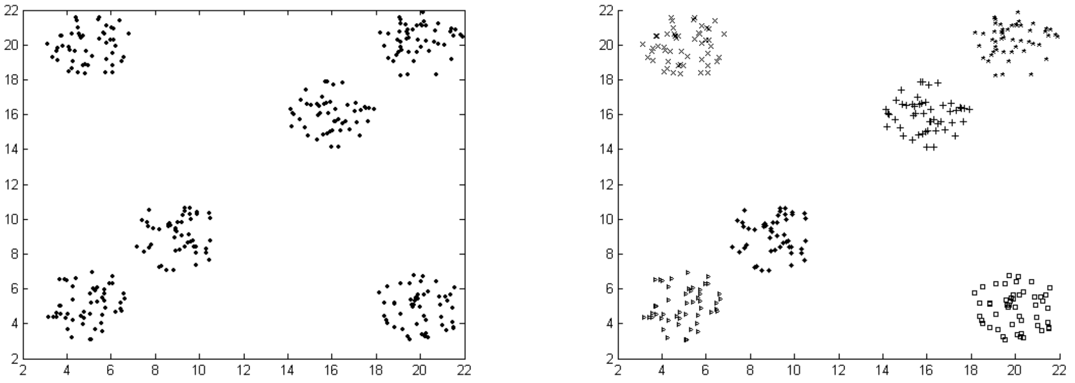
Figure 3. Clustering result for artificial dataset_2.



(a) The unlabelled dataset_3

(b) Clustering with DEMO

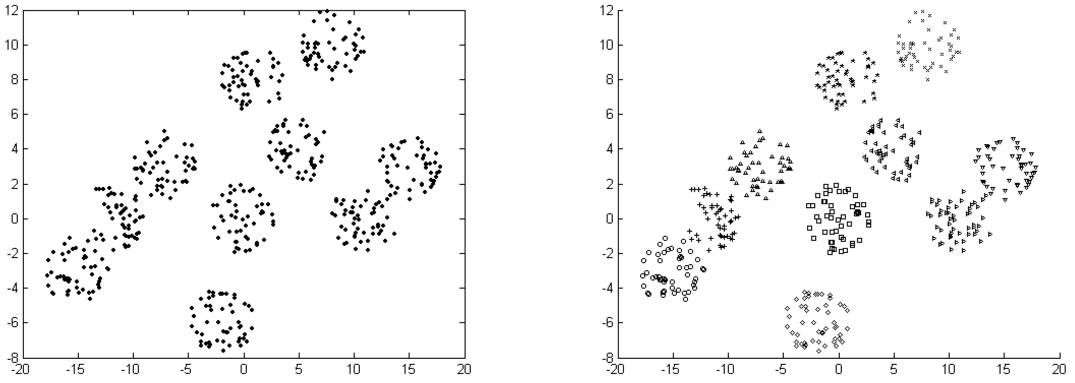
Figure 4. Clustering result for artificial dataset_3.



(a) The unlabelled dataset_4

(b) Clustering with MODE

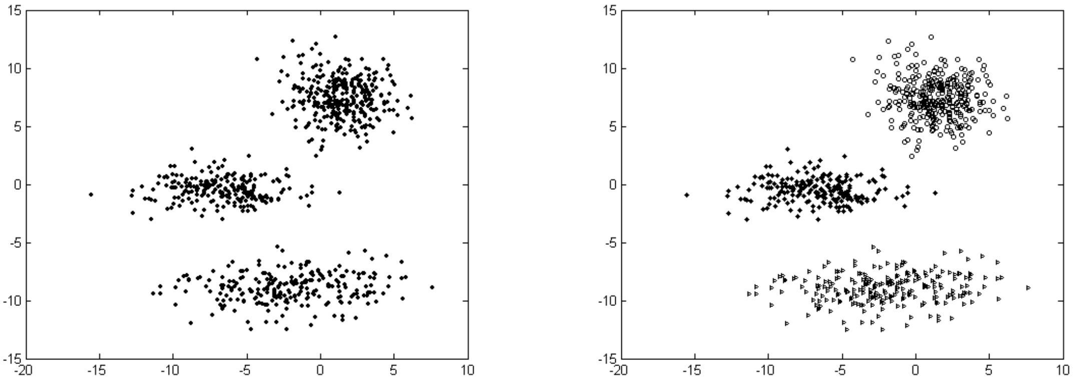
Figure 5. Clustering result for artificial dataset_4.



(a) The unlabelled dataset_5

(b) Clustering with DEMO

Figure 6. Clustering result for artificial dataset_5.



(a) The unlabelled dataset_6

(b) Clustering with PDE

Figure 7. Clustering result for artificial dataset_6.

is drawn consisting of n parallel lines, typically vertical and equally spaced. 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 8. The blue polylines indicate the member genes within a cluster while the black polyline indicates the centroid of that gene. Cluster profile plots (Figure 8) 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.

In Heatmap (Eisen plot), the expression value of a gene at a specific time point is represented by coloring the corresponding cell of the data matrix with a color similar to the original color of its spot on the microarray. The shades of red color represent higher expression level, the shades of green color represent low expression level and the colors towards black represent absence of differential expression values. In our representation, the genes are ordered before plotting so that the genes that belong to the same cluster are placed one after another. The cluster boundaries are identified by white colored blank rows. Figure 9 shows the Heatmap of the seven clusters generated by one run of the MODE algorithm for yeast sporulation data. It is evident from the figure that the expression profiles of the genes of a cluster are similar to each other and they produce similar color patterns.

Studying the functional annotations of a gene within a cluster provides a meaningful way to asses the biological significance of a cluster. Genes within the same cluster are expected to exhibit similar expressions as they should have similar functionality or contribute to the same biological processes. Here we attempt to determine the biological meanings of the clusters by using Gene Ontology (GO) terms using the popular web-based tool FatiGO [33] (`http://www.fatigo.org`) FatiGO extracts the GO terms for a query and a reference set of genes and further computes various statistics for the query set. In our experiment, a query is the set of genes of a cluster and union of the genes from the other clusters is taken as the reference set. The GO level is fixed at three. It is not possible to evaluate each cluster of the final solutions provided by all the algorithms here. So, two interesting clusters from the clustering results obtained on Yeast Sporulation data set by the best performing algorithm (MODE in this case) is examined. Figure 10 shows a part of the FatiGO results of cluster 2 and 6 of multi-objective clustering on the sporulation data. It can be observed that the percentage of genes in the query cluster is considerably different from that of the reference cluster in almost all the functionalities. This implies that the correct genes are selected to remain in the same cluster.

5. Conclusions

This paper compared and contrasted the performance of four state-of-the-art multi-objective variants of DE with two other prominent multi-objective clustering algorithms. The multi-objective DE-variants used the same chromosome representation scheme. Our test-suite included six hand-crafted and four real-life datasets including the gene expression data of budding yeast. The artificial datasets were chosen in two and three dimensions for the ease of visualization of clustering results and the number of clusters for them ranged from 3 to 9. The DE-variants and NSGA-II used the same objective functions based on the Xie-beni index and the FCM index. 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 in terms of both adjusted Rand index and Sillhouette index when all the algorithms were let run for an equal number of generations. Not only did they find out the correct partitions in the data but also in all cases they were able to determine an optimal number of classes with minimum standard deviations. Visualization of the yeast sporulation data clustering results with parallel coordinates and heatmap plots indicate that the MODE yielded compact and well separated clusters. Biological interpretations to the clustering solution have been given with the help of gene annotation using a web-based Gene Ontology tool (FatiGO). Our experimental results indicate that DE is suitable for the multi-objective clustering of a wide variety of datasets. 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 quality of the multi-objective solutions have to be defined. Treating the number of clusters as an objective and thus optimizing the same is also a very important issue for future investigation and this will lead to the fully automatic clustering of previously unhandled datasets with multi-objective DE.

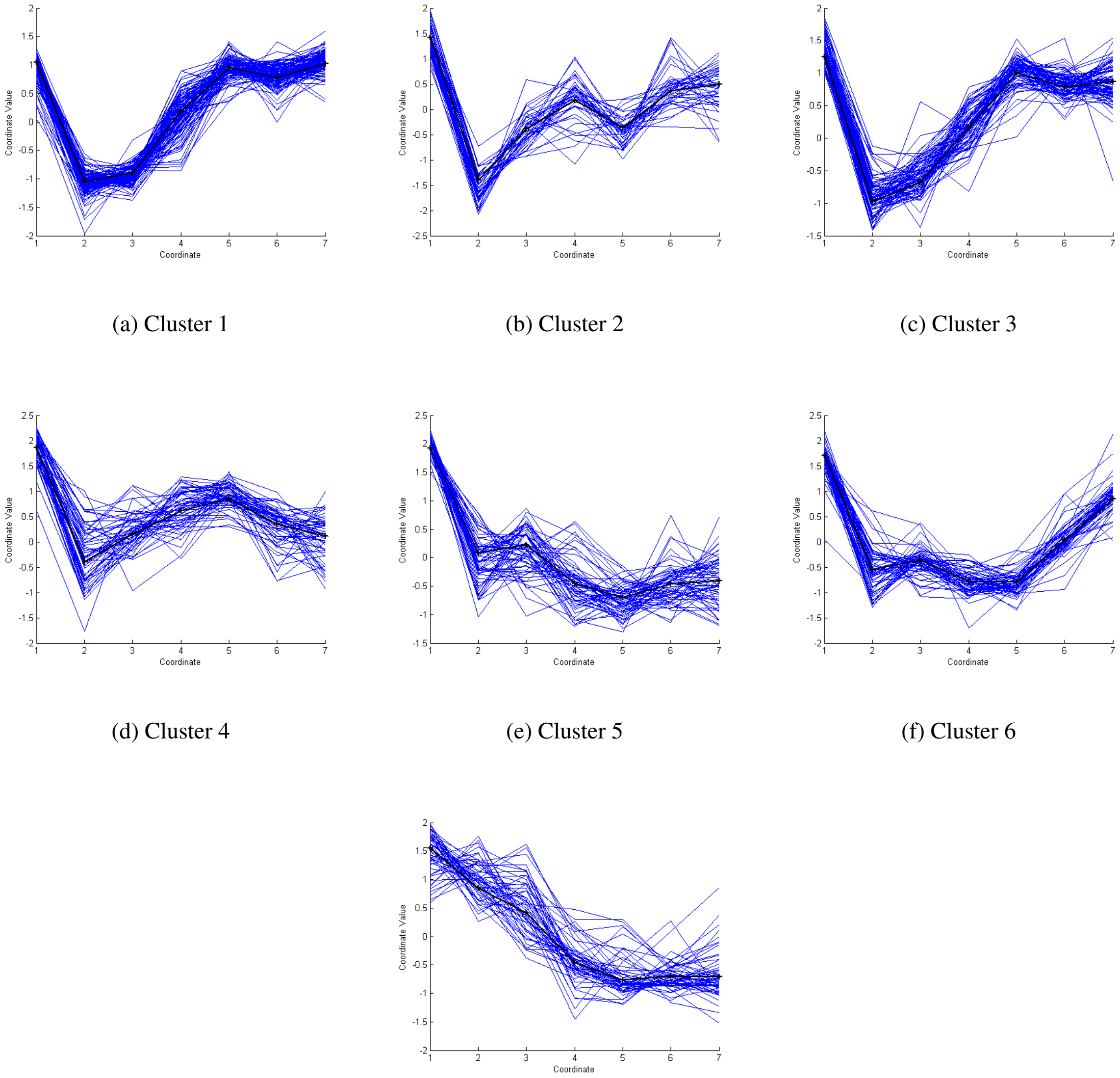
References

[1] 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.

[2] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the Pareto archived evolution strategy”, Evolutionary Computation, vol. 8, no. 2, pp. 149–172, 2000.

[3] R. Storn, K. V. Price, and J. Lampinen, Differential Evolution - A Practical Approach to Global Optimization, Springer, Berlin, 2005.

[4] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons, 2001.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

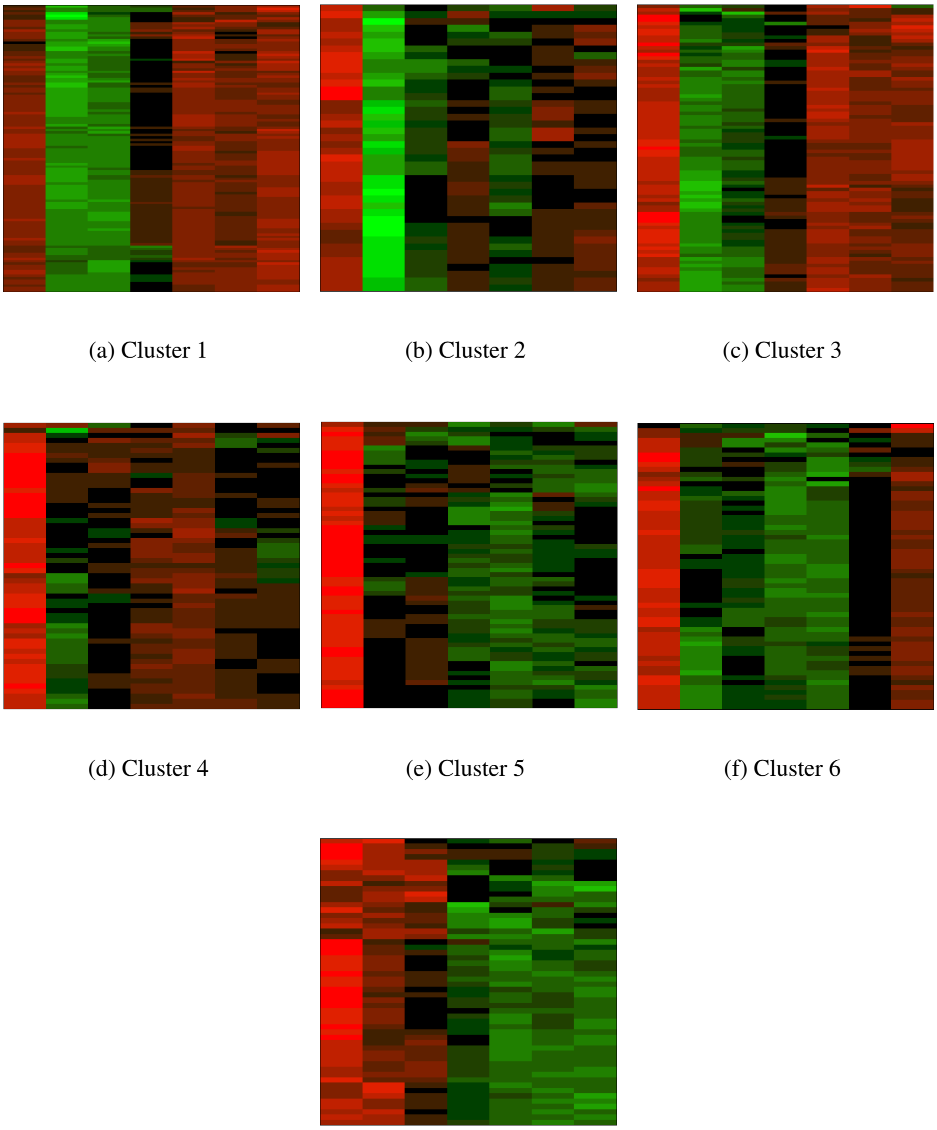
(d) Cluster 4

(e) Cluster 5

(f) Cluster 6

(g) Cluster 7

Figure 8. Cluster profile plots for clustering solution obtained by MODE-based clustering algorithm for yeast sporulation data.



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

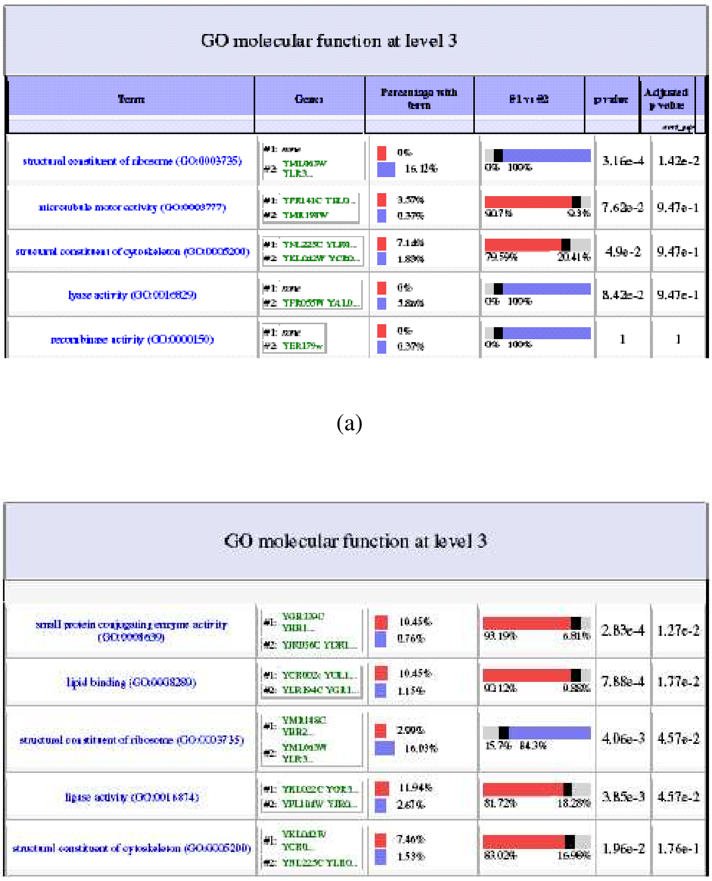
(d) Cluster 4

(e) Cluster 5

(f) Cluster 6

(g) Cluster 7

Figure 9. Heatmaps (Eisen plots) for clustering solution obtained by MODE-based clustering algorithm for yeast sporulation data.



GO molecular function at level 3

Term	Genes	Percentage with term	#1 vs #2	p-value	Adjusted p-value
structural constituent of ribosome (GO:0003735)	#1: none #2: YML063W YLR3...	🟥 0% 🟦 16.12%	0% 100%	3.16e-4	1.42e-2
microtubule motor activity (GO:0003777)	#1: YPR141C YIL0... #2: YMR198W	🟥 3.57% 🟦 0.37%	90.71% 9.3%	7.62e-2	9.47e-1
structural constituent of cytoskeleton (GO:0005200)	#1: YNL225C YLR0... #2: YKL042W YCR0...	🟥 7.14% 🟦 1.89%	79.59% 20.41%	4.9e-2	9.47e-1
lyase activity (GO:0016829)	#1: none #2: YFR055W YAL0...	🟥 0% 🟦 5.86%	0% 100%	8.42e-2	9.47e-1
recombinase activity (GO:0000150)	#1: none #2: YER179w	🟥 0% 🟦 0.37%	0% 100%	1	1

(a)

GO molecular function at level 3

small protein conjugating enzyme activity (GO:0008639)	#1: YOR339C YBR1... #2: YDR059C YDR1...	🟥 10.45% 🟦 0.76%	93.19% 6.81%	2.83e-4	1.27e-2
lipid binding (GO:0008289)	#1: YCR002C YOL1... #2: YLR184C YGR1...	🟥 10.45% 🟦 1.15%	93.12% 6.88%	7.88e-4	1.77e-2
structural constituent of ribosome (GO:0003735)	#1: YMR148C YBR2... #2: YML063W YLR3...	🟥 2.99% 🟦 16.03%	15.7% 84.3%	4.06e-3	4.57e-2
ligase activity (GO:0016874)	#1: YKL022C YOR3... #2: YPL160W YJR0...	🟥 11.94% 🟦 2.67%	81.72% 18.28%	3.85e-3	4.57e-2
structural constituent of cytoskeleton (GO:0005200)	#1: YKL042W YCR0... #2: YNL225C YLR0...	🟥 7.46% 🟦 1.53%	83.02% 16.98%	1.96e-2	1.76e-1

(b)

Figure 10. Part of FatiGO result for (a) cluster 6 and (b) cluster 2 of the best multi-objective clustering algorithm on yeast sporulation dataset.

[5] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007.

[6] A. Abraham, L. C. Jain and R. Goldberg (Eds.), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, Springer Verlag, London, 2005.

[7] H. A. Abbass and R. Sarker, "The Pareto differential evolution algorithm", International Journal on Artificial Intelligence Tools, 11(4):531–552, 2002.

[8] 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.

[9] 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.

[10] 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.

[11] X. Xie and G. Beni, "Validity measure for fuzzy clustering", IEEE Trans. Pattern Anal. Machine Learning, Vol. 3, pp. 841–846, (1991).

[12] J. C. Bezdek, "Cluster validity with fuzzy sets", Journal of Cybernetics, (3) 58–72, (1974).

[13] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic," J. Royal Statist. Soc.: SeriesB (Statistical Methodology), vol. 63, no. 2, pp. 411–423, 2001.

[14] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering", IEEE Transactions on Evolutionary Computation, 11(1):56–76, 2007.

[15] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, Multiobjective genetic clustering for pixel classification in remote sensing imagery, IEEE Transactions Geoscience and Remote Sensing, 2006.

[16] S. Chu et al. "The transcriptional program of sporulation in budding yeast", Science, 282, 699–705, 1998.

[17] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", in M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Julian Merelo, and H.-P. Schwefel, editors, Proceedings of the Parallel Problem Solving from Nature VI Conference, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[18] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", IEEE Transactions on Evolutionary Computation, 3(4):257–271, November 1999.

[19] E. Zitzler and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", IEEE Transactions on Evolutionary Computation, 3(4):257–271, November 1999.

[20] H. A. Abbass, "The self-adaptive Pareto differential evolution algorithm", In Congress on Evolutionary Computation (CEC'2002), volume 1, pages 831–836, Piscataway, New Jersey, IEEE Service Center, 2002.

[21] I. Das, "On characterizing the "knee' of the Pareto curve based on normal-boundary intersection," Structural Optim., vol. 18, no. 2–3, pp. 107–115, 1999.

[22] 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.

[23] K. Deb, "Multi-objective evolutionary algorithms: Introducing bias among Pareto-optimal solutions," in Advances in Evolutionary Computing: Theory and Applications. London, U.K.: Springer-Verlag, pp. 263–292, 2003.

[24] 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.

[25] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the Gap statistic," J. Royal Statist. Soc.: SeriesB (Statistical Methodology), vol. 63, no. 2, pp. 411–423, 2001.

[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] L. Hubert and P. Arabie, "Comparing partitions", Journal of Classification, 193–218, 1985.

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

[30] The Gene Ontology Consortium (2000): "Gene ontology: tool for the unification of biology", Nat. Genet., 25, 25–29.

[31] 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.

[32] M.B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns", Proc. Natl Acad. Sci. USA, 95, 14863–14868, 1998.

[33] F. Al-Shahrour, R. Dłaz-Uriarte, and J. Dopazo, "FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes", Bioinformatics, 20, 578–580, 2004.

[34] http:/www.ics.uci.edu/~mlearn/MLrepository.html

[35] http:/cmgm.stanford.edu/pbrown/sporulation

[36] http://dbkgroup.org/handl/mock/