

Article

Multi-Objective Differential Evolution for Automatic Clustering with Application to Micro-Array Data Analysis

Kaushik Suresh ¹, Debarati Kundu ¹, Sayan Ghosh ¹, Swagatam Das ¹, Ajith Abraham ² and Sang Yong Han ^{3,*}

¹ Dept. of Electronics and Telecommunication Engg, Jadavpur University, Kolkata, India; E-Mails: kaushik_s1988@yahoo.com; kundu.debarati@gmail.com; sayan88tito@gmail.com; swagatamdas19@yahoo.co.in

² Norwegian University of Science and Technology, Norway; E-Mail: ajith.abraham@ieee.org

³ School of Computer Science and Engineering Chung-Ang University, Seoul, Korea

* Author to whom correspondence should be addressed; E-Mail: hansy@cau.ac.kr; Tel.: +82-2-820-5327; Fax: +82-2-825-6996

Received: 1 April 2009; in revised form: 19 May 2009 / Accepted: 22 May 2009 /

Published: 25 May 2009

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 two 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 the multi-objective 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 using six artificial and four real life datasets of varying range of complexities indicate that DE holds immense promise as a candidate algorithm for devising MO clustering schemes.

Keywords: differential evolution; multi-objective optimization; fuzzy clustering; micro-array data clustering

1. Introduction

Optimization-based automatic clustering algorithms greatly rely on a cluster validity function (optimization criterion) whose optima appear as proxies for the unknown “correct classification” in a previously unhandled dataset [1]. Different formulations of the clustering problem vary according to 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 in considerable observable discrepancies between the solutions produced by different algorithms on the same dataset. 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 might 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 little research has been undertaken so far 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 Envelope based Selection) II [13]; both algorithms are multi-objective variants of the 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 two most representative multi-objective DE algorithms on the multi-objective fuzzy clustering problem. The multi-objective DE-variants considered here are namely the Multi-objective DE (MODE) [6] and DE for Multi-objective Optimization (DEMO) [7] owing to their promising results over many benchmark multi-objective optimization problems. Since DE, by nature, is a real-coded population-based optimization algorithm, we here resort to a 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_q) [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 to 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 some microarray yeast sporulation data [16].

2. 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 y :

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

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:

$$\begin{aligned} \forall i \in \{1, \dots, n\}: & \quad f_i(\vec{A}) \geq f_i(\vec{B}) \\ \wedge \exists j \in \{1, \dots, n\}: & \quad f_j(\vec{A}) > f_j(\vec{B}) \end{aligned} \quad (2)$$

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,

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

(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 real-coded representation. It starts with a population of NP real-coded search variable vectors initialized randomly in the feasible search space. 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, to create $\vec{Y}_{i,G}$ for each i -th member, three other parameter vectors (say the r_1 , r_2 , and r_3 -th vectors such that $r_1, r_2, r_3 \in [1, NP]$ 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:

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

for $j = 1, 2, \dots, D$ and $\text{rand}_j(0, 1) \in [0,1]$ 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 $\vec{R}_{i,G}$ gets at least one component from $\vec{Y}_{i,G}$. 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 the next generation $G = G+1$. 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:

$$\vec{Z}_{i,G+1} = \begin{cases} \vec{R}_{i,G} & \text{if } f(\vec{R}_{i,G}) \leq f(\vec{Z}_{i,G}) \\ \vec{Z}_{i,G} & \text{if } f(\vec{R}_{i,G}) > f(\vec{Z}_{i,G}) \end{cases} \quad (7)$$

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

2.3. The Multi-Objective Variants of DE

We consider here the two most promising multi-objective variants of DE: the Multi-Objective DE (MODE) [6] and the DE for Multi-objective Optimization (DEMO) [7]. We briefly discuss here the outline of the algorithms instead of reiterating their details, already available in the cited literature.

1) MODE: MODE was proposed by Xue *et al.* in [6]. 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-dimensional unconstrained problems with 250,000 evaluations and the results are compared only to those obtained by SPEA [19].

2) DEMO: DEMO was proposed by Robic and Filipic [7]. This algorithm 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) [7]. Below we provide a pseudo-code for MODE/parent:

1. Evaluate the initial population \mathbf{P} of random individuals.
2. While stopping criterion not met, do:
 - 2.1. For each individual \vec{X}_i ($i = 1 \dots NP$) from \mathbf{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 \mathbf{P} .

In DEMO the candidate replaces the parent if it dominates it. If the parent dominates the candidate, the candidate is discarded. Otherwise (when the candidate and the parent is non-dominated with regard to each other), the candidate is added to the population. This step is repeated until NP number of candidates is created. After that, we get a population of the size between NP and $2.NP$. If the population has enlarged, we have to truncate it to prepare it for the next step of the algorithm.

The truncation consists of sorting the individuals with non-dominated sorting and then evaluating the individuals of the same front with the crowding distance metric. The truncation procedure keeps in the population only the best NP individuals (with regard to these two metrics). The described

truncation is derived from NSGA-II. DEMO incorporates two crucial mechanisms. The immediate replacement of the parent individual with the candidate that dominates it is the core of DEMO. The newly created candidates that enter the population (either by replacement or by addition) instantly take part in the creation of the following candidates. This emphasizes elitism within reproduction, which helps achieving the first goal of multi objective optimization – convergence to the true Pareto front. The second mechanism is the use of non-dominated sorting and crowding distance metric in truncation of the extended population. Besides preserving elitism, this mechanism stimulates the uniform spread of solutions. This is needed to achieve the second goal – finding as diverse non-dominated solutions as possible. DEMO’s selection scheme thus efficiently pursues both goals of multi objective optimization.

The other two variants were inspired by the concept of Crowding DE as recently introduced by Thomsen [22]. When optimizing functions with many optima, we would sometimes like not only to find one optimal point, but also discover and maintain multiple optima in a single algorithm run. For this purpose, Crowding DE can be used. Crowding DE is basically conventional DE with one important difference. Usually, the candidate is compared to its parent. In Crowding DE, the candidate is compared to the most similar individual in the population. The applied similarity measure is the Euclidean distance between the two solutions.

The second, DEMO/closest/dec, works in the same way as DEMO/parent, with the exception that the candidate solution is compared to the most similar individual in decision space. If it dominates it, the candidate replaces this individual; otherwise it is treated in the same way as in DEMO/parent. The applied similarity measure is the Euclidean distance between the two solutions in decision space. In the third variant, DEMO/closest/obj, the candidate is compared to the most similar individual in objective space. DEMO/closest/dec and DEMO/closest/obj need more time for one step of the procedure than DEMO/parent. This is because at every step they have to search for the most similar individual in the decision and objective space, respectively.

3. Multi-Objective Clustering Scheme

3.1. Search-Variable Representation and Scheme for Finding Correct Number of Clusters

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 real 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:

$$\vec{Z}_{i,G} = \begin{array}{|c|c|c|c|c|c|c|c|} \hline T_{i,1} & T_{i,2} & \dots & T_{i,K_{\max}} & \vec{m}_{i,1} & \vec{m}_{i,2} & \dots & \vec{m}_{i,K_{\max}} \\ \hline \end{array} \quad (8)$$

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:

$$\text{IF } T_{i,j} > 0.5 \text{ THEN the } j\text{-th cluster center } \bar{m}_{i,j} \text{ is ACTIVE ELSE } \bar{m}_{i,j} \text{ is INACTIVE.} \quad (9)$$

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 a penalized version of the FCM function J_q as the two objectives. The FCM measure J_q may be defined as:

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

where q is the fuzzy exponent, d indicates a distance measure between the j -th pattern vector and i -th cluster centroid, k is the number of active cluster centroids 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 σ to the minimum separation sep of the clusters. Here σ and sep may be written as:

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

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

The XB index is then written as:

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

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 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:

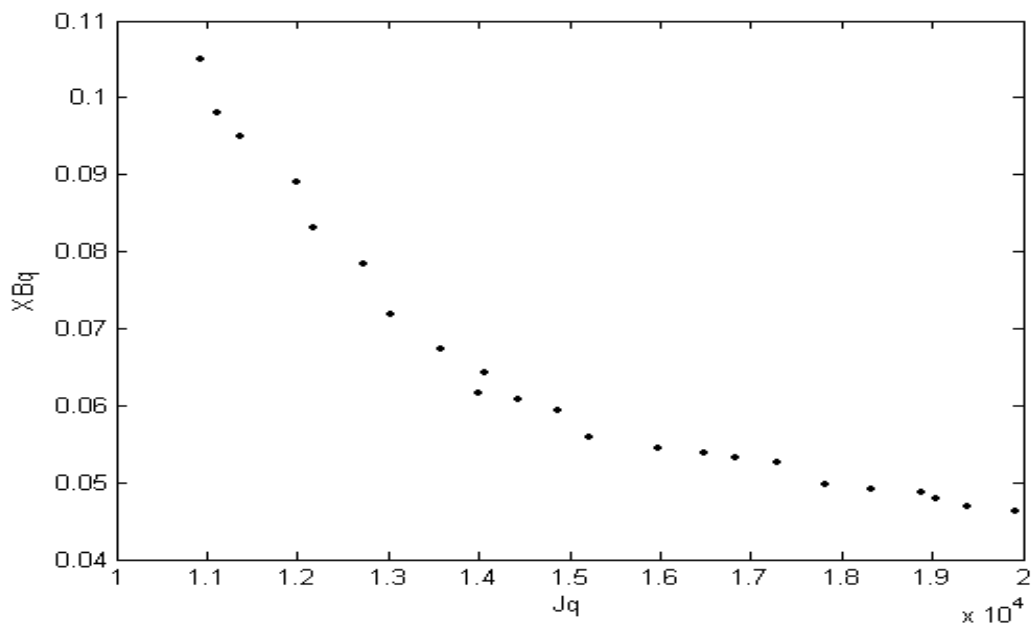
$$u_{ij} = \frac{1}{\sum_{p=1}^k \left(\frac{d(\bar{m}_i, \bar{Z}_j)}{d(\bar{m}_p, \bar{Z}_j)} \right)^{\frac{2}{q-1}}} \quad (14)$$

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 assignment:

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

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_q 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 and micro-array 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.

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



Note that except MOCK, all the DE-based algorithms here use the objective functions described in (10) and (13). The NSGA-II based algorithm described in [3] use a plain FCM index that incorporates no compensation due to large number of clusters. This is obvious, as the method of [3] assumes the number of clusters to be known beforehand, whereas, the multi-objective clustering framework proposed here makes room for a variable number of clusters and the modified FCM index of (10) penalizes a large number of clusters. MOCK also uses two conflicting objective functions known as the

overall deviation and connectivity. The overall deviation is computed as the overall summed distances between data items and their corresponding cluster center:

$$Dev(C) = \sum_{C_k \in C} \sum_{i \in C_k} \delta(i, \bar{m}_k) \quad (16)$$

where C is the set of all clusters, \bar{m}_k is the centroid of cluster C_k , and δ is the chosen distance function (here, the Euclidean distance). As an objective, overall deviation should be minimized.

Again connectedness evaluates the degree to which neighboring data points have been placed in the same cluster. It is computed in the following way:

$$Conn(C) = \sum_{i=1}^n \left(\sum_{j=1}^L x_{i, nn_{ij}} \right) \quad (17)$$

where $x_{r,s} = 1/j$, if there does not exist any C_k such that $r \in C_k \cap s \in C_k$,
 $x_{r,s} = 0$, otherwise.

nn_{ij} is the j -th nearest neighbor of datum i , n is the size of the clustered data set, and L is a parameter determining the number of neighbors that contribute to the connectivity measure. As an objective, connectivity should be minimized.

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 [23, 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 follow a similar technique as the one used in MOCK. It is inspired by Tibshirani *et al.*'s Gap statistic [25], 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. We use the same heuristic technique described in pages 65 – 66 of [4] to generate the attainment scores for each clustering problem. Finally, we plot the attainment scores as a function of the number of clusters. All solutions corresponding to the local optima in the resulting plot are

considered as promising solutions. The global maximum in this plot may be considered as the estimated “best” solution.

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 [26] (which is a generalization of the Rand index [27]) and the silhouette index [28]. Mostly we use the adjusted Rand index for evaluating the quality of partitioning in those 9 datasets for which the nominal classification is known. Silhouette index is used for the Yeast microarray dataset, corresponding to which no standard or nominal classification exists. In most recent and existing literatures, like [3, 29], the clustering quality on yeast sporulation data has been judged by using this index.

The adjusted Rand index comes as a generalization of the Rand Index [27]. 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:

$$R = \frac{\sum_{i,j} \binom{n_{ij}}{2} - [\sum_i \binom{n_{i+}}{2} \cdot \sum_j \binom{n_{+j}}{2}] / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{n_{i+}}{2} + \sum_j \binom{n_{+j}}{2}] - [\sum_i \binom{n_{i+}}{2} \cdot \sum_j \binom{n_{+j}}{2}] / \binom{n}{2}}, \quad (18)$$

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 $n_{+j} = \sum_i n_{ij}$ is the number of objects classified into class j in the actually known classification.

Silhouette width reflects the compactness and separation of the clusters. Given a set of data points $Z = \{\vec{Z}_1, \dots, \vec{Z}_n\}$ and a given clustering solution $C = \{C_1, C_2, \dots, C_k\}$, 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:

$$s(\vec{Z}_j) = \frac{b(\vec{Z}_j) - a(\vec{Z}_j)}{\max(a(\vec{Z}_j), b(\vec{Z}_j))}. \quad (19)$$

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 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 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:

$$s(C) = \frac{1}{n} \sum_{j=1}^n s(\vec{Z}_j) \quad (20)$$

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.

3.6. Putting It Together

Putting the above procedures together, we may now give an over all pseudo-code of the DE-based multi-objective clustering algorithm in the following way:

Pseudo code of clustering with multi-objective DE:

1. Randomly initialize the control genes and cluster centroids for the maximum number of clusters for the initial population. Each control gene corresponds to a cluster centroid. A centroid is said to be “active” if the corresponding gene exceeds 0.5.
2. While stopping criterion not met, do:
 - 2.1. Evaluate values of Xie-Beni and penalized FCM indices.
 - 2.2. Create trial vector of control genes and cluster centroids using standard DE operators and optimize the indices using multi-objective differential evolution. The control genes and cluster centroids are thus evolved.
3. Find the Pareto front of the final set of solutions and find the best solution using gap statistic. The solution at knee point on the Pareto front corresponds to the correct number of clusters.

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, 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 [30] 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 publicly available from the website: <http://cmgm.stanford.edu/pbrown/sporulation>. This dataset consists of 6,118 genes measured across seven 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 6,118 genes, those 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 [31], 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 [32].

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

4.2. Other Competitor Algorithms

This paper compares the clustering performances of two promising multi-objective DE-variants with two other evolutionary multi-objective clustering techniques: the NSGA – II [23] and MOCK [24]. Below we briefly describe these techniques, to provide an idea of their conceptual difference with the DE-based MO clustering algorithms.

1) The NSGA II based Clustering Algorithm : Bandyopadhyay *et al.* [3] proposed a non-automatic multi-objective scheme for clustering the pixels of remote sensing satellite images into several fuzzy partitions. They employed the NSGA II algorithm to optimize a number of fuzzy cluster validity indices simultaneously. In NSGA II, initially a random parent population G_0 of size N is created. Then the population is sorted based on the non-domination relation. Each solution of the population is assigned a fitness that is equal to its non-domination level. A child population H_0 is created from the parent population G_0 by using binary tournament selection, recombination, and mutation operators. Generally according to this algorithm, initially a combined population $R_t = G_t + H_t$ is formed of size R_t , which is $2N$. Now all the solutions of R_t are sorted based on their non-domination status. If the total number of solutions belonging to the best non-dominated set F_1 is smaller than N , F_1 is completely included into G_{t+1} . The remaining members of the population G_{t+1} are chosen from subsequent non-dominated fronts in the order of their ranking. To choose exactly N solutions, the solutions of the last included front are sorted using the crowded comparison operator and the best among them (i.e., those with larger values of the crowding distance) are selected to fill in the available slots in G_{t+1} . The new population G_{t+1} is now used for selection, crossover, and mutation to create a new population H_{t+1} of size N and the process continues. The crowding distance operator is also used in the parent selection phase in order to break a tie in the binary tournament selection. This operator is instrumental in maintaining diversity in the Pareto front.

The resultant set of near-Pareto-optimal solutions contained a number of non-dominated solutions, which the user could judge relatively and pick up the most promising one according to the problem requirements. Real-coded encoding of the cluster centers was used for this purpose. We shall use this

algorithm for clustering synthetic as well as real life datasets with real numerical attributes in this paper.

2) The MOCK Algorithm : Handl and Knowles proposed a multi-objective clustering scheme known as Voronoi Initialized Evolutionary Nearest-Neighbor Algorithm (VIENNA [32]), which is based on PESA II [33] and simultaneously optimizes two objectives. It employs a straightforward encoding of a clustering, with a gene for each data item and its allele value specifying the cluster to which the data item should belong. VIENNA needed an advanced initialization scheme based on Voronoi cells and directed mutation to make up for deficiencies in its encoding. In addition, it is non-automatic and does not provide any means to select good solutions from the final Pareto front. Handl and Knowles [4, 34] proposed an improved EMO-based clustering algorithm, which they named *Multi-Objective Clustering with Automatic k Determination* (MOCK). They fine-tuned one of the objectives used in VIENNA and found a better encoding that does not fix the number of clusters and because of good locality and heritability, allows a much more effective exploration of the search space via suitable operators. They also developed a method for selection of best solutions from the Pareto front based on a null model, thus also determining the number of clusters, automatically. MOCK was also extended in [35] for improving its scalability to large, high-dimensional datasets and data with large number of clusters. Handle and Knowles also introduced MOCK-around-medoids, which allows for the clustering of similarity data [36] (as opposed to vectorial data, i.e. points in a metric space). Here we shall use for comparison the version of the algorithm described in [4].

4.3. 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 and Cr was fixed at 0.9. These parameter values have been recommended for DE after performing a series of hand-tuning experiments. First we use standard values of F (0.8) and Cr (0.9) [21] and repeat the clustering techniques on various datasets with varying population size NP . We find that keeping NP around 40 gives reasonable computational time over a wide range of datasets. Next, fixing NP at 40 we varied F and Cr respectively and obtained the clustering results on several datasets in terms of the adjusted Rand index. Figure 2 shows a glimpse of these experiments with MODE on artificial datasets 5 and 6 for various values of F , keeping Cr at 0.9. In each case we report the average adjusted Rand index for 30 independent runs of the algorithms. Similarly Figure 3 presents the final accuracy of MODE on the same two datasets for various values of Cr keeping the value of F random between 0.5 and 1.5. It is evident from both the figures that the suggested parameter setting (F random and $Cr = 0.9$) gives best clustering performance with the multi-objective DE variants. We do not provide the results for all the available datasets to save space and considering the fact that they show more or less similar trend as shown in Figures 2 and 3.

Figure 2. Final clustering result for artificial datasets 5 and 6 with MODE for different settings of scale factor F .

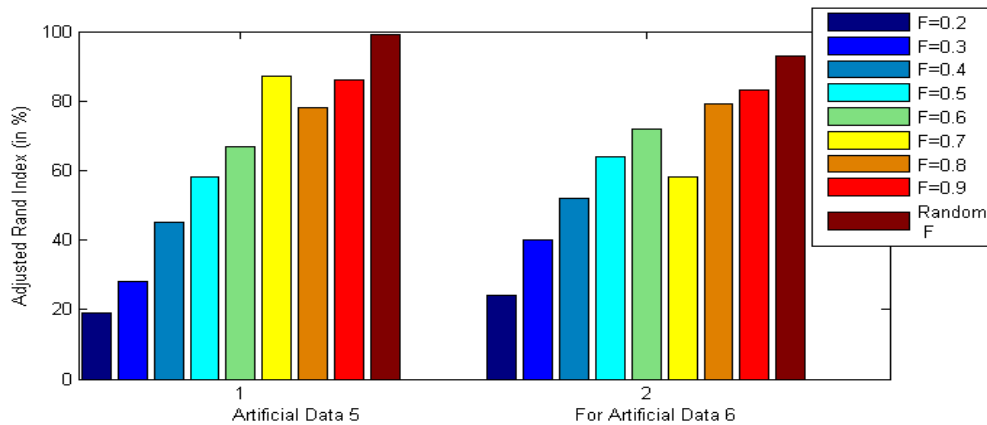
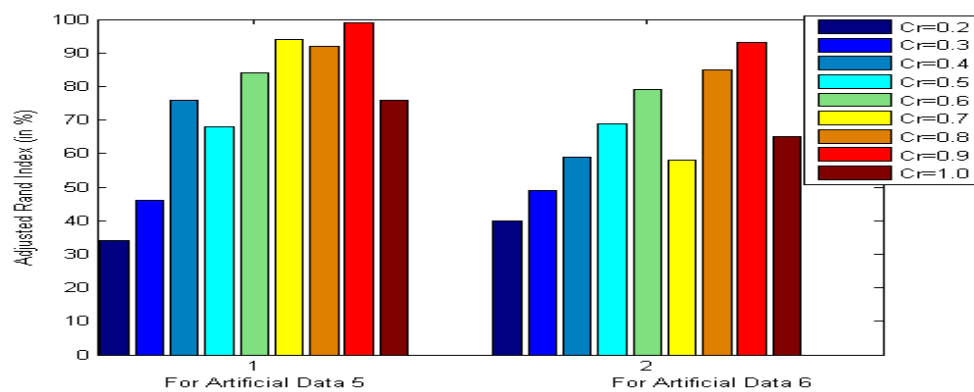


Figure 3. Final clustering result for artificial datasets 5 and 6 with MODE for different settings of crossover rate Cr .



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 two DE variants and the NSGA II use the same parameter representation scheme. Clustering with MOCK was performed with the source codes available from <http://dbkgroup.org/handl/mock/>.

4.4. Presentation of Results

The mean adjusted Rand index values of the best-of-run solutions provided by six contestant algorithms over the nine datasets (for which the nominal classifications are already known) have been provided in Table 2. The best entries have been marked in boldface in each row. Note that for the DE-based MO clustering techniques, the number of estimated classes correspond to the best solution from the Pareto optimal front chosen by using the technique described in Section 3.4. Table 3 shows the Silhouette index values for yeast sporulation data as no standard nominal classification is known for this dataset. Note that we have not provided the results for other datasets in terms of Silhouette index

as analysis based on the Silhouette width is not an objective evaluation technique, as it may be biased towards algorithms optimizing objectives related to the Silhouette width.

Tables 4 and 5 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 adjusted Rand index and Silhouette index. For all cases in Tables 4 and 5, sample size = 30 and number of degrees of freedom = 58. Here all the t -tests have been performed using the statistical calculator available from the website: <http://www.graphpad.com/quickcalcs/ttest1.cfm>

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 two multi-objective DE variants) have been depicted in Figures 4 to 9.

4.5. 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) and the heatmap plot in MATLAB 7.0.4 version. Parallel coordinates [37] 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 10. The blue polylines indicate the member genes within a cluster while the black polyline indicates the centroid of that gene. Cluster profile plots (Figure 10) 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 (aka Eisen plot) [38], 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. Figure 11 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 colour patterns. 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 [39] (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 12 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.

Table 2. Mean value of adjusted Rand index found and standard deviations (in parentheses) by four contestant algorithms over 30 independent runs on nine datasets.

Dataset	Algorithms							
	MODE DEM		O		NSGA2		MOCK	
	<i>k</i>	Adjusted Rand Index	<i>k</i>	Adjusted Rand Index	<i>k</i>	Adjusted Rand Index	<i>k</i>	Adjusted Rand Index
Dataset_1	9.12 (1.46)	0.846199 (0.031257)	9.43 (0.843)	0.828437 (0.046182)	9.37 (1.72)	0.802180 (0.004782)	8.52 (2.81)	0.810934 (0.0059348)
Dataset_2	3.36 (0.65)	0.957621 (0.006312)	3.74 (0.363)	0.9273464 (0.0008573)	3.16 (0.072)	0.9378123 (0.006821)	3.33 (1.03)	0.946547 (0.004536)
Dataset_3	4.14 (0.36)	0.951786 (0.004827)	4.09 (0.24)	1.000000	3.57 (0.51)	0.963841 (0.0046719)	3.78 (1.25)	0.878732 (0.0712523)
Dataset_4	6.04 (0.25)	1.000000	6.13 (1.27)	0.857463 (0.065639)	6.28 (0.46)	0.957818 (0.004678)	6.08 (0.51)	0.978761 (0.006734)
Dataset_5	9.24 (3.89)	0.983785 (0.076764)	10.03 (0.37)	0.993173 (0.089371)	12.43 (0.939)	0.947641 (0.006646)	10.41 (0.80)	0.9454568 (0.0012043)
Dataset_6	5.19 (0.93)	0.93456 (0.08463)	5.62 (0.867)	0.881136 (0.078348)	4.65 (1.58)	0.881395 (0.056483)	5.16 (0.38)	0.910294 (0.016743)
Iris	3.04 (0.16)	0.738626 (0.0756779)	2.98 (0.40)	0.748784 (0.067457)	2.16 (1.06)	0.715898 (0.005739)	3.05 (0.37)	0.736574 (0.075763)
Wine	3.16 (0.46)	0.875849 (0.0087642)	3.65 (0.83)	0.858876 (0.0035287)	3.88 (0.67)	0.828645 (0.0074653)	3.59 (0.46)	0.864764 (0.0034398)
Breast Cancer	2.08 (0.38)	0.956456 (0.0056453)	2.68 (0.64)	0.912173 (0.0043247)	2.57 (0.60)	0.944236 (0.006521)	2.10 (0.53)	0.9465731 (0.006748)

Table 3. Average Silhouette index and number of clusters found and standard deviations (in parentheses) by four contestant algorithms over 30 independent runs on the Yeast sporulation dataset.

Dataset	Algorithms							
	MODE DEMO		NSGA2		MOCK			
	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index	<i>k</i>	Silhouette Index
Yeast Sporulation	7.08 (0.12)	0.676434 (0.00072)	6.34 (0.32)	0.558619 (0.057832)	7.22 (0.68)	0.641306 (0.04813)	6.67 (0.857)	0.613567 (0.005738)

Table 4. Unpaired *t*-test Results for adjusted Rand index.

Dataset Std.	Err	<i>t</i>	95% Conf. Intvl	Two-tailed <i>P</i> Stat	tical Significance Level
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	6.3744	-0.0309 to 0.0059	0.1774	Extremely Significant
Wine	0.003	2.3999	-0.0118 to -0.0010	0.0278	Significant
Breast Cancer	0.009	1.3744	-0.0309 to 0.0059	0.1774	Not Significant

Table 5. Unpaired *t*-test results for Silhouette index

Dataset Std.	Err	<i>t</i>	95% Conf. Intvl	Two-tailed <i>P</i> -Values	Statistical Significance Level
Yeast Sporulation	0.003	2.3999	-0.0118 to -0.0010	0.0214	significant

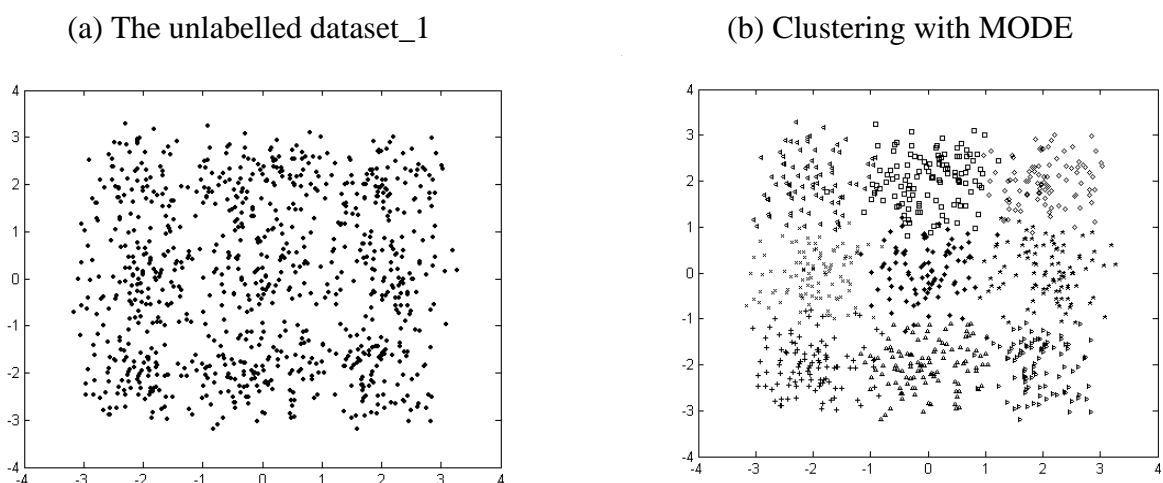
Figure 4. Clustering result for artificial dataset_1.

Figure 5. Clustering result for artificial dataset_2.

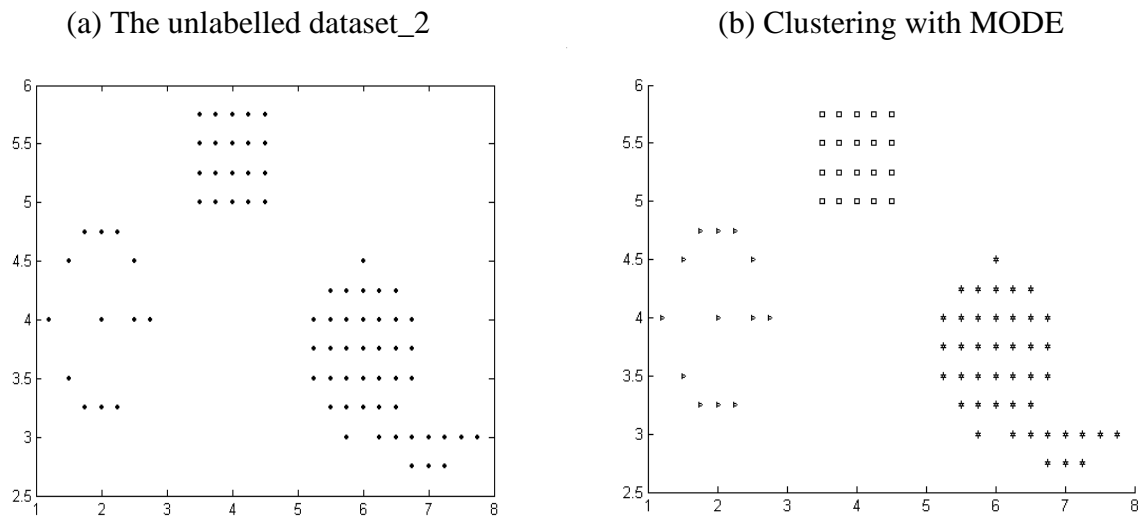


Figure 6. Clustering result for artificial dataset_3.

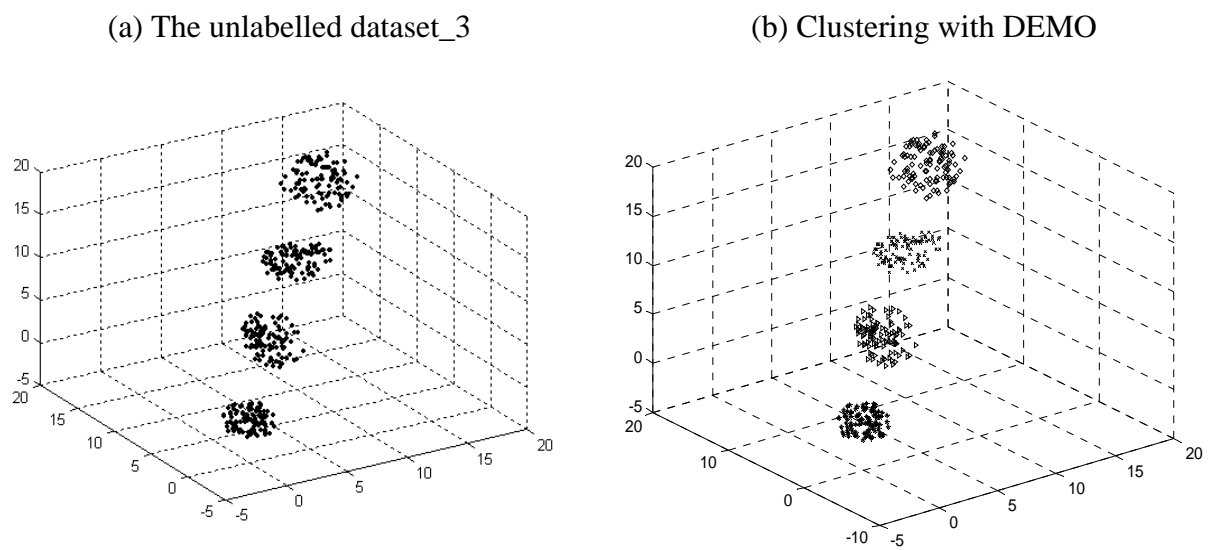


Figure 7. Clustering result for artificial dataset_4.

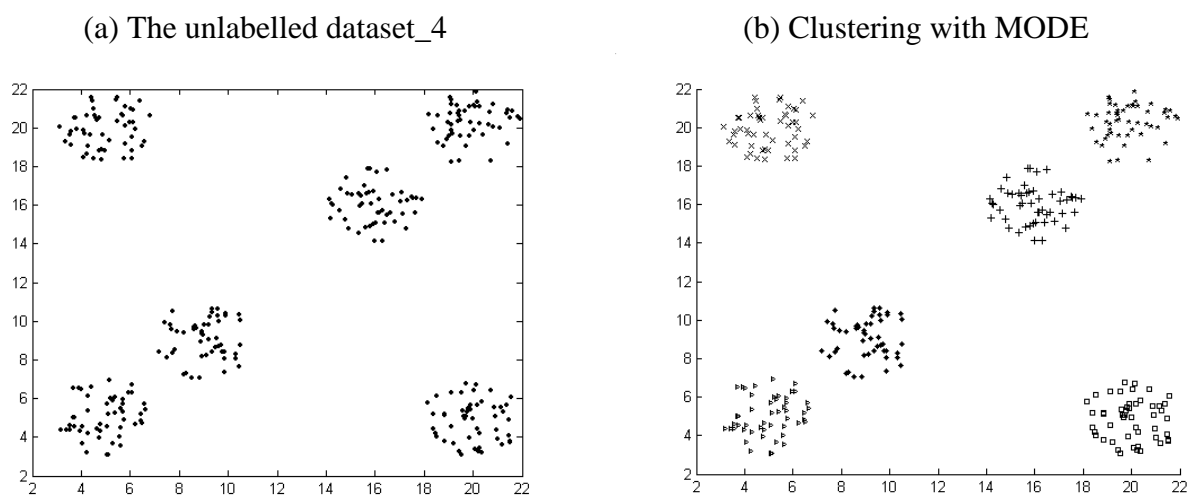


Figure 8. Clustering result for artificial dataset_5.

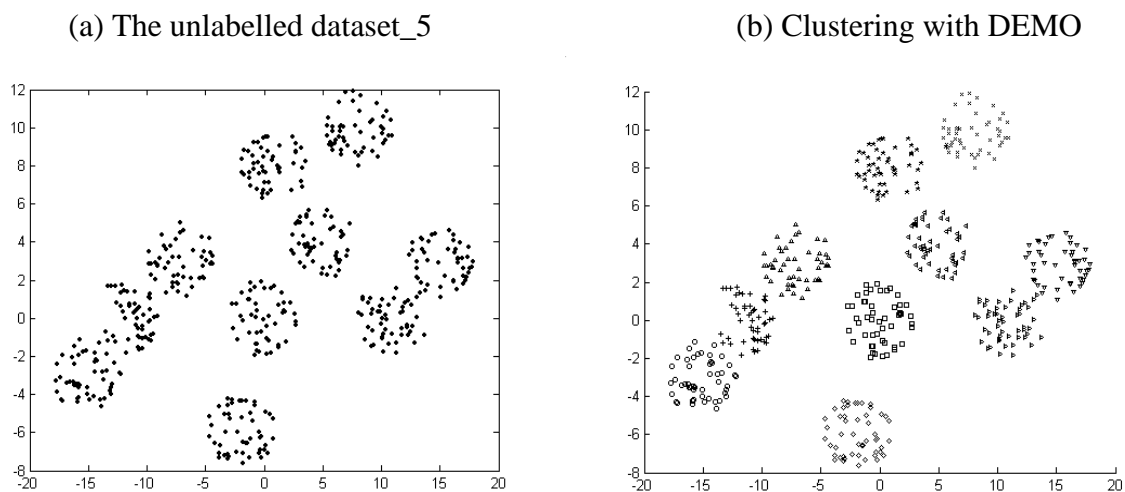


Figure 9. Clustering result for artificial dataset_6.

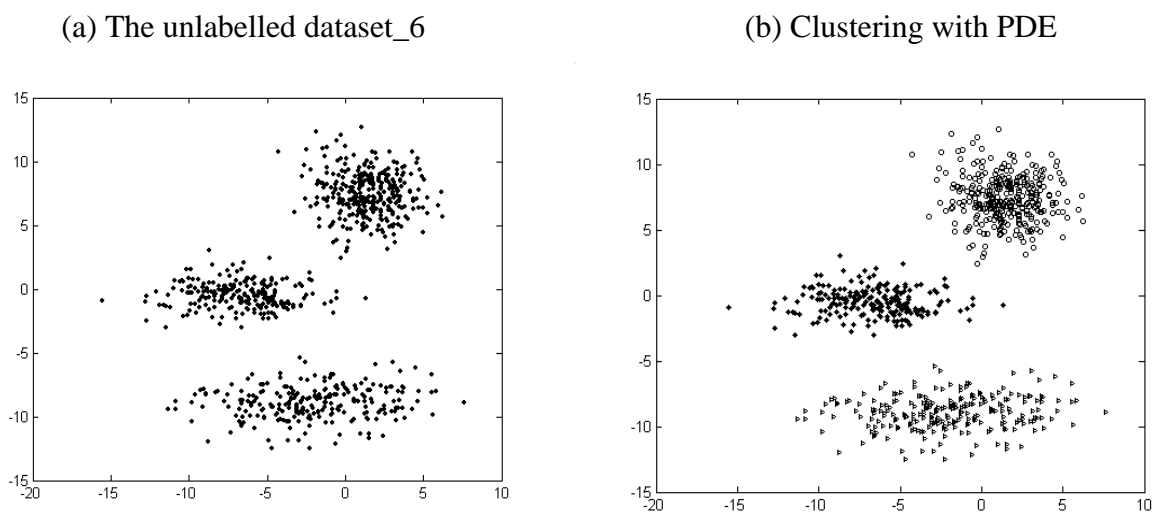


Figure 10. Cluster profile plots for clustering solution obtained by MODE-based clustering algorithm for yeast sporulation data.

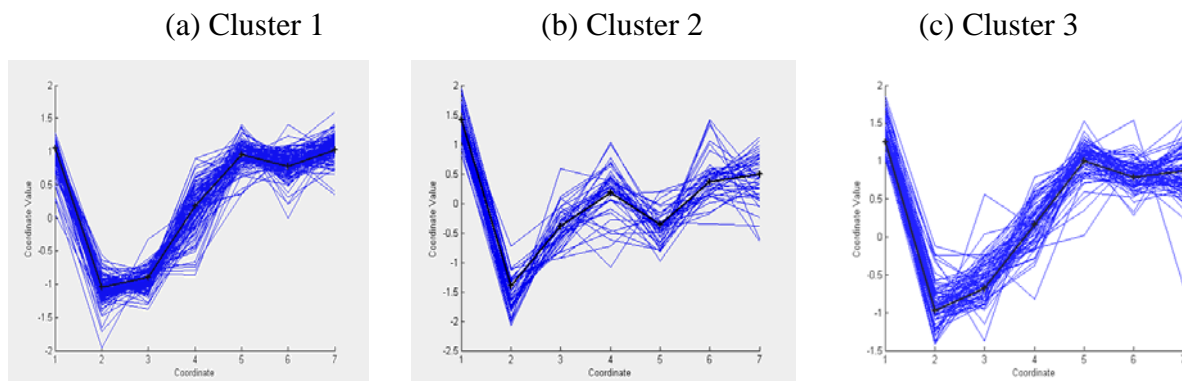


Figure 10. Cont.

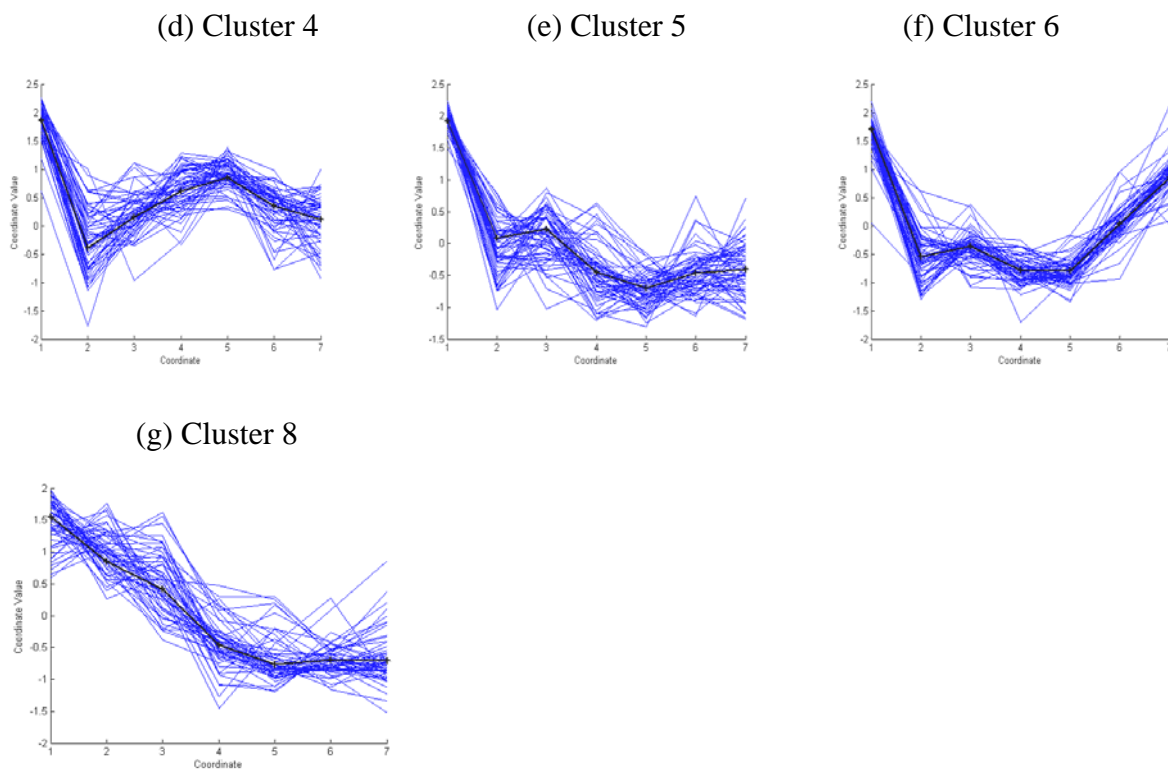


Figure 11. Heatmaps (Eisen plots) for clustering solution obtained by MODE-based clustering algorithm for yeast sporulation data.

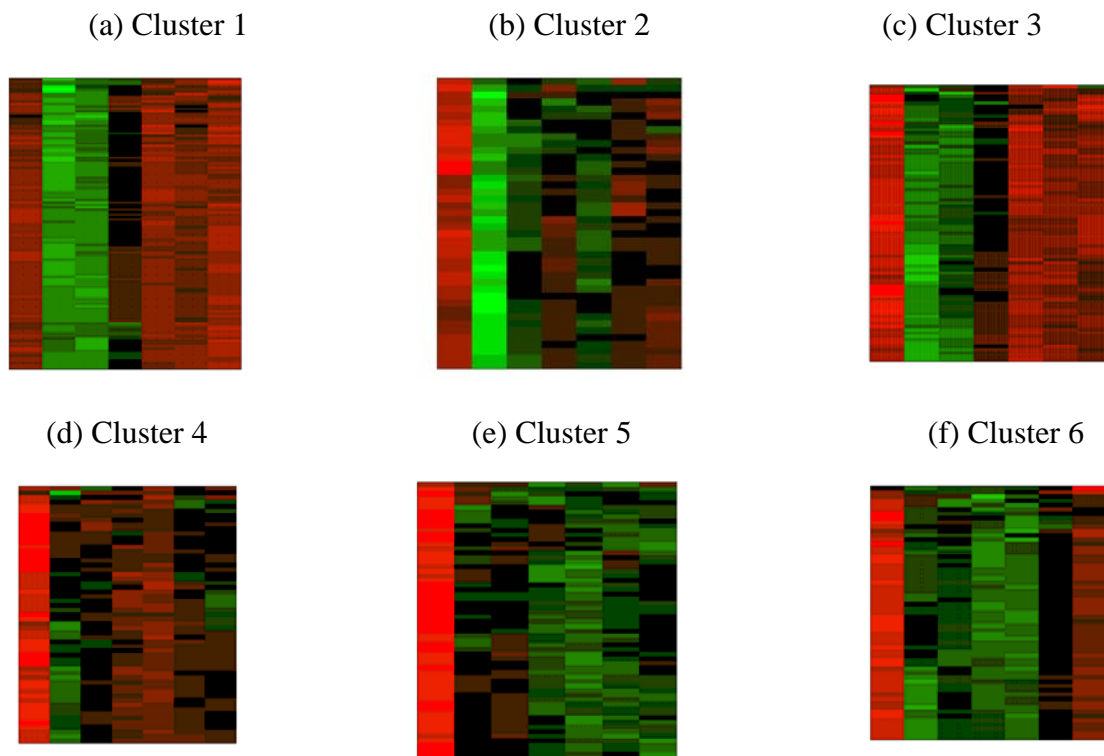


Figure 11. Cont.

(g) Cluster 7

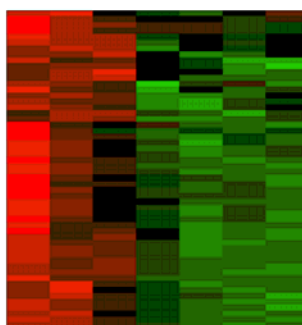


Figure 12. Part of FatiGO result for (a) cluster 6 and (b) cluster 2 of the best multi-objective clustering algorithm on yeast sporulation dataset.

(a)

GO molecular function at level 3					
Term	Genes	Percentage with term	#1 vs #2	p value	Adjusted p value
					sort_up
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 YEL0... #2: YMR198W	3.57% 0.37%	90.7% 9.3%	7.62e-2	9.47e-1
structural constituent of cytoskeleton (GO:0005200)	#1: YNL225C YLR0... #2: YKL042W YCR0...	7.14% 1.83%	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

(b)

GO molecular function at level 3					
small protein conjugating enzyme activity (GO:0008639)	#1: YOR339C YHR1... #2: YJR036C YDR1...	10.45% 0.76%	93.19% 6.81%	2.83e-4	1.27e-2
lipid binding (GO:0008289)	#1: YCR002c YOLL... #2: YLR194C YGR1...	10.45% 1.15%	90.12% 9.88%	7.88e-4	1.77e-2
structural constituent of ribosome (GO:0003735)	#1: YMR188C 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: YPL104W 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

5. Conclusions

This article compared the performances of two state-of-the-art multi-objective variants of DE with two other prominent multi-objective clustering algorithms. The 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 Silhouette 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). 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. The multi-objective clustering framework may be utilized for various real life applications, such as offline classification of sensor data, automatic image segmentation, document clustering etc.

Acknowledgement

This research was supported in parts by (a) the Ministry of Knowledge of Economy, Korea, under the Chung-Ang University HNRC-ITRC (Home Network Research Center) support program supervised by the Institute of Information Technology Assessment and (b) a grant(CR070019) from Seoul R&BD Program funded by the Seoul Development Institute of Korean government

References and Notes

1. Jain, A.K.; Murty, M.N.; Flynn, P.J. Data clustering: a review. *ACM Comput. Surv.* **1999**, *31*, 264-323.
2. Xu, R.; Wunsch, D. Clustering. *Series on Computational Intelligence*; IEEE Press: Piscataway, NJ, USA, 2008.
3. Bandyopadhyay, S.; Maulik, U.; Mukhopadhyay, A. Multiobjective genetic clustering for pixel classification in remote sensing imagery. *IEEE Trans. Geosci. Remote Sens.* **2006**, *41*, 1075-1081.
4. Handl J.; Knowles, J. An evolutionary approach to multiobjective clustering. *IEEE Trans. Evol. Comput.* **2007**, *11*, 56-76.

5. Abbass, H.A.; Sarker, R. The pareto differential evolution algorithm. *Int. J. Artif. Intell. Tools* **2002**, *11*, 531-552.
6. Xue, F.; Sanderson, A.C.; Graves, R. J. Pareto-based multi-objective differential evolution. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, Canberra, Australia, 2003; Volume 2, pp. 862-869.
7. Robic, T.; Filipic, B. DEMO: Differential Evolution for Multiobjective Optimization, In *Evolutionary Multi-Criterion Optimization, Third International Conference, EMO 2005* Coello Coello, C.A., Aguirre, A.H., Zitzler, E., Eds.; Springer Lecture Notes in Computer Science: Guanajuato, Mexico, 2005; Vol. 3410, pp. 520-533;
8. Iorio, A.W.; Li, X. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence, Proceedings*; Springer-Verlag, Lecture Notes in Artificial Intelligence: Berlin, Germany, 2004; Vol. 3339, pp. 861-872.
9. Paterlinia, S.; Krink, T. Differential evolution and particle swarm optimisation in partitional clustering. *Comput. Stat. Data Anal.* **2006**, *50*, 1220-1247.
10. Omran, M.; Engelbrecht, A.P.; Salman, A. Differential evolution methods for unsupervised image classification, In *Proceedings of Seventh Congress on Evolutionary Computation (CEC-2005)*, 2005.
11. Das, S.; Abraham, A.; Konar, A. Automatic clustering using an improved differential evolution algorithm. *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans* **2008**, *38*, 1-20.
12. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182-197
13. Corne, D.W.; Knowles, J.D.; Oates, M.J. The pareto-envelope based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature - PPSN VI*, Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., Schwefel, H-P., Eds.; Springer Lecture Notes in Computer Science: Berlin, Germany, 2000; 869-878.
14. Xie, X.; Beni, G. Validity measure for fuzzy clustering. *IEEE Trans. Pattern Anal. Machine Intelligence* **1991**, *3*, 841-846.
15. Bezdek, J.C., Cluster validity with fuzzy sets. *J. Cybern.* **1974**, 58-72.
16. Chu, S.; DeRisi, J.; Eisen, M.; Mulholland, J.; Botstein, D.; Brown, P.O.; Herskowitz, I. The transcriptional program of sporulation in budding yeast. *Science* **1998**, *282*, 699-705.
17. Sawaragi, Y.; Nakayama, H.; Tanino, T. Theory of multiobjective optimization. In *Mathematics in Science and Engineering*; Academic Press Inc.: Orlando, FL, USA, 1985; Vol. 176.
18. Deb, K. *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons: Chichester, UK, 2001.
19. Coello Coello, C.A.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers: New York, USA, 2007.
20. Storn, R.; Price, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **1997**, *11*, 341–359.
21. Storn, R.; Price, K.V.; Lampinen, J. *Differential Evolution - A Practical Approach to Global Optimization*; Springer: Berlin, Germany, 2005.

22. Thomsen, R. Multimodal optimization using crowding-based differential evolution. In *Congress on Evolutionary Computation (CEC'2004)*, Portland, OR, USA, IEEE Service Center, 2004; Volume 1, pp. 1382-1389.
23. Mattson, C.A.; Mullur, A.A.; Messac, A. Smart Pareto filter: Obtaining a minimal representation of multiobjective design space. *Eng. Optim.* **2004**, *36*, 721-740.
24. Branke, J.; Deb, K.; Dierolf, H.; Osswald, M. Finding knees in multi-objective optimization. In *Proc. 8th Int. Conf. Parallel Problem Solving From Nature*, 2004; pp. 722-731.
25. Tibshirani, R.; Walther, G.; Hastie, T. Estimating the number of clusters in a dataset via the Gap statistic. *J. Royal Statist. Soc.: Ser. B (Stat. Method.)* **2001**, *63*, 411-423.
26. Hubert, L.; Arabie, P. Comparing partitions. *J. Classific.* **1985**, 193-218.
27. Rand, W.M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **1971**, *66*, 846-850.
28. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, *J. Comput. Appl. Math.* **1987**, *20*, 53-65.
29. Bandyopadhyay, S.; Mukhopadhyay, A.; Maulik, U. An improved algorithm for clustering gene expression data. In *Bioinformatics*; Oxford University Press: Oxford, UK, 2007; vol. 23, no. 21, pp. 2859-2865.
30. Blake, C.; Keough, E.; Merz, C.J. UCI repository of machine learning database, 1998. Available online: <http://www.ics.uci.edu/~mllearn/MLrepository.html/>.
31. Theodoridis, S.; Koutroumbas, K. *Pattern Recognition*, Second Edition; Elsevier Academic Press: San Diego, CA, USA, 2003.
32. Handl, J.; Knowles, J.; Evolutionary multiobjective clustering. In *Proc. 8th Int. Conf. Parallel Problem Solving from Nature*, 2004; pp. 1081-1091.
33. Corne, D. W.; Jerram, N. R.; Knowles, J. D.; Oates, M. J.; PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proc. Genetic Evol. Comput. Conf.*, 2001; pp. 283-290.
34. Handl, J.; Knowles, J. Exploiting the tradeoff—the benefits of multiple objectives in data clustering. In *Proc. 3rd Int. Conf. Evol. Multicriterion Optim.*, 2005; pp. 547-560.
35. Handl, J.; Knowles, J. Improvements to the scalability of multiobjective clustering. In *Proc. IEEE Congr. Evol. Comput.*, 2005; Vol. 3, pp. 2372-2379.
36. Handl, J.; Knowles, J.; Multiobjective clustering around medoids. In *Proc. IEEE Congr. Evol. Comput.*, 2005; vol. 1, pp. 632-639.
37. Keim, D.A.; Kriegel, H.P. Visualization techniques for mining large databases: a comparison. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 923-938.
38. Eisen, M.B.; Spellman, P.T.; Brown, P.O.; Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci* **1998**, *95*, 14863-14868.
39. Al-Shahrour, F.; Díaz-Uriarte, R.; Dopazo, J.; FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* **2004**, *20*, 578-580.