Integrating Multi-Objective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering

Yimin Liu, Tansel Özyer, Reda Alhajj and Ken Barker
Advanced Database Systems and Applications Lab
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
{liuyi, ozyer, alhajj, barker}@cpsc.ucalgary.ca

Keywords: multi-objective genetic algorithm, clustering, k-means, validity analysis

Received: March 15, 2004

Clustering algorithms in general need the number of clusters as a priori, which is mostly hard for domain experts to estimate. In this paper, we use Niched Pareto k-means Genetic Algorithm (GA) for clustering. After running the multi-objective GA, we get the pareto-optimal front that gives the optimal number of clusters as a solution set. We analyze the clustering results using several cluster validity techniques proposed in the literature, namely Silhoutte, C index, Dunn's index, DB index, SD index and S-Dbw index. This gives an idea about ranking the optimal number of clusters for each validity index. We demonstrate the applicability and effectiveness of the proposed clustering approach by conducting experiments using two datasets: Iris and the well-known Ruspini dataset.

Povzetek: "[Click here and Enter short Abstract in Slovene language]"

1 Introduction

Data mining methods and techniques have been successfully applied to different areas including bioinformatics. They are designed for extracting previously unknown significant relationships and regularities out of huge heaps of details in large data collections [11].

Classification is one of the well-known mining techniques. It has two main aspects: discrimination and clustering. In discrimination analysis, also known as supervised clustering, observations are known to belong to pre-specified classes. The task is to allocate predictors for the new coming instances to be able to classify them correctly. In contrast to classification, in clustering, also known as unsupervised clustering, classes are unknown *a prior*; the task is to determine classes from the data instances. Clustering is used to describe methods to group unlabeled data. By clustering, we aim to discover gene/samples groups that enable us to discover, for example, the functional role or the existence of a regulatory novel gene among the members in a group. The literature shows that increasing attention is devoted to the development of new clustering techniques [12]. Existing clustering techniques mostly used for gene expression data clustering can be classified into traditional clustering algorithms including hierarchical clustering [20], partitioning [22], and recently emerging clustering techniques such as graph-based [19] and model-based [21, 23] approaches.

As described in the literature, some of the existing clustering techniques have been successfully employed in analyzing gene expression data. These include hierarchical clustering, partitional clustering, graph-based clustering, and model-based clustering. In general, existing clustering techniques require pre-specification of the number of clusters, which is not an easy task to predict *a prior* even for experts. Thus, the problem handled in this paper may be identified as follows: Given a set of data instances, we mainly concentrate on microarray data, it is required to develop an approach that produces different alternative solutions, and then conduct validity analysis on the resulting solutions to rank them.

Different assumptions and terminologies were considered for the components of the clustering process and the context in which clustering is used. There exist fuzzy clustering techniques as well as hard clustering techniques. Hard clustering assigns a label l_i to each object x_i, identifying its class label. The set of all labels for the object set is $\{ l_1 , l_2 ,\ldots, l_n \}$, where $l_i \in \{ l_1 , l_2 ,\ldots, l_k \}$, and k is the number of clusters. In fuzzy clustering, an object may belong to more than one cluster, but with different degree of membership; an object x_i is assigned to cluster j based on the value of the corresponding function f_{ij}. The membership of an object may not be precisely defined; there is likelihood that each object may or may not be member of some clusters. The presence of noise in the data set may be quite high.

Our approach presented in this paper has been designed to smoothly handle the clustering of different data sets. In the existing approaches, the number of clusters is mostly given a-priori. This motivated us to consider the idea of proposing multi-objective k-means

genetic algorithm (MOKGA) approach in order to present to the user several alternatives without taking the weight values into account. Otherwise, the user will have several trials weighting with different values until a satisfactory result is obtained. We evaluate the obtained candidate optimal number of clusters by applying the cluster validity techniques, namely Silhoutte, C index, Dunn's index, DB index, SD index and S-Dbw index. Finally, the proposed approach has been tested using the Iris and Ruspini datasets.

As *K*-Means clustering is concerned, it is a commonly used algorithm for partition clustering [22]. It is a widely used technique and has been utilized to analyze gene expression data. The purpose of *K*-Means clustering is the optimization of an objective function that is described by the equation:

$$E = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, m_i) \quad (1)$$

where m_i is the center of cluster C_i, and $d(x, m_i)$ is the Euclidean distance between a point x and m_i. It can be seen that the criterion function attempts to minimize the distance between each point and the center of its cluster. The algorithm begins by randomly initializing a set of C cluster centers, then assigns each object of the dataset to the cluster whose center is the nearest, and re-computes the centers. This process is repeated until the total error criterion converges.

The rest of the paper is as follows. Section 2 is an overview of the multi-objective approach. Section 3 describes the proposed system: namely, clustering and cluster validity analysis. Section 4 includes the experimental results. Section 5 is the conclusions.

2 Multi-objective Genetic Algorithms

A multi-objective optimization problem has *n* decision variables, *k* objective functions, and *m* constraints. Objective functions and constraints are functions of the decision variables. The optimization goal may be described as follows:

$$\begin{aligned} &\max imize \setminus \min imize \quad y = f(x) = (f_1(x), f_2(x), ..., f_k(x)) \\ &subject \quad to \quad e(x) = (e_1(x), e_2(x), ..., e_m(x)) \le 0 \\ &where \qquad x = ((x_1, x_2, ..., x_n) \in X \\ &\qquad\quad y = ((y_1, y_2, ..., y_n) \in Y \end{aligned} \quad (2)$$

where x is the decision vector, y is the objective vector, X denotes the decision space, and Y is called the objective space. The constraints $e(x) \ge 0$ determine the set of feasible solutions [14].

Solutions to a multi-objective optimization method are mathematically expressed in terms of non-dominated or superior points. In a minimization problem, a vector $x^{(1)}$ is partially less than another vector $x^{(2)}$, denoted $x^{(1)} \prec x^{(2)}$, when no value of $x^{(2)}$ is less than $x^{(1)}$ and at least one value of $x^{(2)}$ is strictly greater than $x^{(1)}$. If $x^{(1)}$ is partially less than $x^{(2)}$, we say that $x^{(1)}$ dominates $x^{(2)}$ or the solution $x^{(2)}$ is inferior to $x^{(1)}$. Any vector which is not dominated by any other vectors is said to be non-dominated or non-inferior. The optimal solutions to a multi-objective optimization problem are non-dominated solutions [13].

A common difficulty with the multi-objective optimization is the conflict between the objective functions. None of the feasible solutions allows optimal solutions for all the objectives. Pareto-optimal is the solution, which offers the least objective conflict. In traditional multi-objective optimization, multiple objectives are combined to form one objective function. One of the traditional methods being used is weighting each objective and scalarizing the result. At the end of each run, pareto-optimal front may be obtained, which actually represents one single point.

3 The Proposed Approach

In this paper, we propose a new clustering approach, namely Multi-Objective Genetic K-means algorithm (MOKGA), which is a general purpose approach for clustering other datasets after modifying the fitness functions and changing the proximity values as distance or non-decreasing similarity function according to the requirements of the dataset to be clustered.

Concerning our approach, after running the multi-objective k-means genetic algorithm, we get the pareto-optimal front that gives the optimal number of clusters as a solution set. Then, the system analyzes the clustering results found under six of the cluster validity techniques proposed in the literature, namely Silhoutte, *C* index, Dunn's index, SD index, DB index and S_Dbw index.

3.1 A. Multi-Objective Genetic K-Means Algorithm

The Multi-Objective Genetic K-means Algorithm (MOKGA) is basically the combination of the Fast Genetic K-means Algorithm (FGKA) [1] and Niched Pareto Genetic Algorithm [2].

As presented in the flowchart shown in Figure 1, MOKGA uses a list of parameters to drive the evaluation procedure as in the other genetic types of algorithms: including population size (number of chromosomes), t_dom (number of comparison set) representing the assumed non-dominated set, crossover, mutation probability and the number of iterations that the execution of the algorithm needs to obtain the result.

Sub-goals can be defined as fitness functions; and instead of scalarizing them to find the goal as the overall fitness function with the user defined weight values, we expect the system to find the set of best solutions, i.e., the pareto-optimal front. By using the specified formulas, at each generation, each chromosome in the population is evaluated and assigned a value for each fitness function.

The coding of our individual population is a chromosome of length *n*. Each allele in the chromosome takes a value from the set {1, 2, …, *K*}, and represents a pattern. The value indicates the cluster that the corresponding pattern belongs to. Each chromosome exhibits a solution set in the population. If the chromosome has *k* clusters, then each gene a_n (n=1 to N) takes different values from [1..*k*].

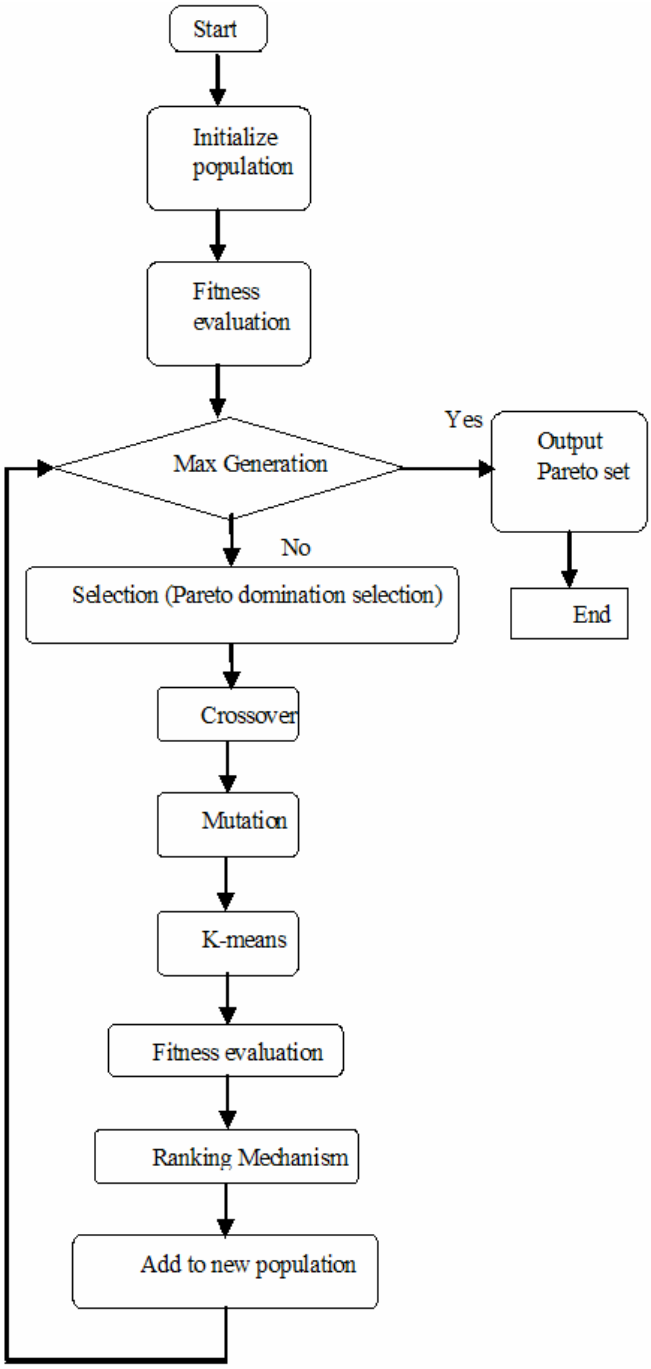

Figure 1: Flow chart: the process of the Multi-Objective Genetic K-means Algorithm

Initially, *current generation* is assigned to zero. Each chromosome takes *number of clusters* parameter within the range 1 to maximum number of clusters given by the user. A population with the specified number of chromosomes is created randomly by using the method described in [5]: Data points are randomly assigned to each cluster at the beginning; then the rest of the points are randomly assigned to clusters. By using this method, we can avoid generating illegal strings, which means some clusters do not have any pattern in the string.

Using the current population, the next population is generated and generation number is incremented by 1. During the next generation, the current population performs the pareto domination tournament to get rid of the worst solutions from the population, crossover, mutation and k-means operator [1] to reorganize each object's assigned cluster number. Finally, we will have twice the number of individuals after the pareto domination tournament. We apply the ranking mechanism used in [15] to satisfy the elitism and diversity preservation. By using this method the number of individuals is halved.

The first step in the construction of the next generation is the selection using pareto domination tournaments: In this step, two candidate items picked among (*population size- t_{dom}*) individuals participate in the pareto domination tournament against the t_{dom} individuals for the survival of each in the population. In the selection part, t_{dom} individuals are randomly picked from the population. With two randomly selected chromosome candidates in (*population size- t_{dom}*) individuals, each of the candidates is compared against each individual in the comparison set, t_{dom}. If one candidate has a larger total within-cluster variation fitness value and a larger number of cluster values than of all of the chromosomes in the comparison set, this means it is dominated by the comparison set already and will be deleted from the population permanently. Otherwise, it resides in the population.

After the pareto domination tournament, one-point crossover operator is applied on randomly chosen two chromosomes. The crossover operation is carried out on the population with the crossover p_c. After the crossover, assigned cluster number for each gene is renumbered beginning from a_1 to a_n. For example, if two chromosomes having 3 clusters and 5 clusters, respectively, need to have a crossover at the third location:

Number of clusters=3: 1 2 3 3 3
Number of clusters=5: 1 4 3 2 5

We will get 1 2 3 2 5 and 1 4 3 3 3; and then they are renumbered to get the new number of clusters parameters:

Number of clusters=4: 1 2 3 2 4 (for 1 2 3 2 5)
Number of clusters=3: 1 2 3 3 3 (for 1 4 3 3 3)

The reason for choosing one-point crossover is because it produced better results compared to multi-point after some initial experiments.

The mutation operator on the current population is employed after the crossover. During the mutation, we replace each gene value a_n by a_n' with respect to the probability distribution; for n=1, ..., N simultaneously. a_n' is a cluster number randomly selected from {1, …, K} with the probability distribution $\{p_1, p_2, ..., p_K\}$ defined using the following formula:

$$p_i = \frac{1.5 * d_{\max}(X_n) - d(X_n, c_k)}{\sum_{k=1}^{K} 1.5 * d_{\max}(X_n) - d(X_n, c_k)} \quad (3)$$

where $i \in [1..k]$ and $d(X_n, C_k)$ denotes Euclidean distance between pattern X_n and the centroid C_k of the k-th cluster. $d_{\max}(X_n) = \max_k\{d(X_n, C_k)\}$, p_i represents what

the probability interval of mutating gene is assigned to cluster *i* (e.g., Roulette Wheel).

Finally, k-means operator is applied. It is used to reanalyze each chromosome gene's assigned cluster value; it calculates the cluster centre for each cluster; and then it re-assigns each gene to the cluster that is the closest one to the instance in the gene. Hence, *k*-means operator is used to speed up the convergence process by replacing a_n by a_n' for n=1, ..., N simultaneously, where a_n' is the closest to object X_n in Euclidean distance.

After all the operators are applied, we have twice the number of individuals, after having the pareto dominated tournament. We can not give an exact number as equal to the number of initial population size because at each generation randomly picked candidates are picked for the survival test leading to deletion of one or both, in case dominated. To halve the number of individuals, having the number of individuals we had, the ranking mechanism proposed in [15] is employed. So, the individuals obtained after crossover, mutation and k-means operator are ranked, and we pick the best individuals among them to place in the population for the next generation.

Our approach picks the first *l* individuals considering the elitism and diversity among *2l* individuals. Pareto fronts are ranked. Basically, we find the pareto-optimal front and remove the individuals of the pareto-optimal front from *2l* set and place it in the population to be run in the next generation. In the remaining set, again we get the first pareto-optimal front and we put it in the population and so on. Since we try to get the first *l* individuals, the last pareto-optimal front may have more individuals required to complete the number of individuals to *l*, we handle the diversity automatically. We rank them and reduce the objective dimension into one. Then, we sum the normalized value of the objective functions of each individual. We sort them in increasing order and find each individual's total difference from its individual pairs, the one with the closest smaller summed values and the one with the closest greater summed values. After sorting the individuals in terms of each one's total difference in decreasing order, we keep placing from the top as many individuals as we need to complete the number of population to *l*. The reason for doing this is to take the crowding factor into account automatically, so that individuals occurring closer to others are unlikely to be picked. Solutions far apart from the others will be considered for the necessity of diversity. Further details are given in [15]. This method was also suggested as a solution for the elitism and diversity for improvement in NSGA-II.

Finally, if the maximum number of generations is reached, or the prespecified threshold is satisfied then exit; otherwise the next generation is performed.

During our clustering process, we defined two objective functions: minimizing the number of clusters and minimizing the partitioning error. To partition the *N* pattern points into *K* clusters one goal is to minimize the Total Within-Cluster Variation (*TWCV*), which is specified as:

$$TWCV = \sum_{n=1}^{N}\sum_{d=1}^{D} X_{nd}^2 - \sum_{k=1}^{K}\frac{1}{Z_k}\sum_{d=1}^{D} SF_{kd}^2 \qquad (4)$$

where $X_1, X_2, .., X_N$ are the N objects, X_{nd} denotes feature d of pattern X_n (n = 1 to N). SF_{kd} is the sum of the d-th features of all the patterns in cluster k (G_k) and Z_k denote the number of patterns in cluster k (G_k) and SF_{kd} is:

$$SF_{kd} = \sum_{\overrightarrow{X_n} \in G_k} X_{nd}, \quad (d = 1, 2, ...D). \qquad (5)$$

And the other objective function is to minimize the *number of clusters* parameter. Under the lights of these two objective functions, after running the algorithm, we aim at obtaining the first pareto optimal front having the best partition with the least number of clusters as optimal solution set.

3.2 Cluster Validity Techniques

Clustering is an unsupervised task and after clustering the data, partitioning into subgroups, we need to check its validity. The criteria widely accepted by the clustering algorithms are the compactness of the cluster and their well-separateness. Those criteria should be validated and optimal clusters should be found, so the correct input parameters must be given to the satisfaction of optimal clusters. Basically, the number of clusters is given as a priori. However, pareto-optimal solution set for the clustering results is obtained in our approach, MOKGA. We believe that these are the good clustering outcomes, and we use the cluster validity index to decide and see the overall picture of those validity index value changes for each *number of clusters* parameter value in the solution set. In our system, we considered six cluster validity techniques widely used for the validation task. These are Dunn index [4], Davies-Bouldin index [3], Silhouette index [5], C index [6], SD index [8] and S_Dbw index [9]. Based on the validated results, the optimal number of clusters can be determined.

The SD validity index definition is based on the concepts of average scattering for clusters and total separation between clusters. The average scattering for clusters is defined as:

$$Scatt(n_c) = \frac{1}{n_c}\sum_{i=1}^{n_c} \left\| \sigma(v_i) \right\| \Big/ \left\| \sigma(x) \right\| \qquad (6)$$

where $\sigma(v_i)$ is the average standard deviation (average of the Euclidian distance between all the points) of cluster centers; and $\sigma(x)$ is the average standard deviation of all the data points. The total separation between clusters is defined as:

$$Dis(n_c) = \frac{D_{\max}}{D_{\min}} \sum_{k=1}^{n_c} \left(\sum_{i=1}^{n_c} \left\| v_k - v_z \right\| \right)^{-1} \qquad (8)$$

where, D_{max} = max($\|v_i - v_j\|$) $\forall i, j \in \{1, 2,3,\ldots, n_c\}$ is the maximum distance between cluster centers and D_{min}= min($\|v_i - v_j\|$) $\forall i, j \in \{1, 2,\ldots, n_c\}$ is the minimum distance between cluster centers.

The SD index is calculated using the following equation:

$$SD(n_c) = \alpha \times Scat(n_c) + Dis(n_c) \qquad (9)$$

where α is a weighting factor.

In the above equation, Scat(n_c) indicates the average compactness of clusters. A small value for this term indicates compact clusters. *Dis*(n_c) indicates the total separation between the *n* clusters. Since the two terms of *SD* have different ranges, a weighting factor is needed to incorporate both terms in a balanced way. The number of clusters that minimizes the index is an optimal value.

S_Dbw is formalized based on the clusters' compactness (intra-cluster variance) and the density (Inter-cluster Density) between clusters. Inter-cluster density is defined as follows:

$$Dens\_bw(n_c) = \frac{1}{n_c \times (n_c - 1)} \sum_{i=1}^{n_c} \left(\sum_{\substack{j=1 \\ i \neq j}}^{n_c} \frac{density(u_{ij})}{\max\{density(v_i), density(v_j)\}} \right) \quad (10)$$

where v_i and v_j are centers of clusters c_i and c_j; and u_{ij} is the middle point of the line segment defined by the clusters' centers v_i and v_j. The term *density(u)* is given by following equation:

$$density(u) = \sum_{l=1}^{n_{ij}} f(x_l, u) \quad (11)$$

where n_{ij} is the number of tuples that belong to the cluster c_i and c_j, i.e., $x_l \in c_i$, and $c_j \in S$. Function $f(x,u)$ is defined as:

$$f(x,u) = \begin{cases} 0, & if \quad d(x,u) > stedev \\ 1, & otherwise \end{cases} \quad (12)$$

where *stedev* is the average standard deviation of clusters.

Inter-cluster Density (ID) evaluates the average density in the region among clusters in relation to the density of the clusters. Intra-cluster variance measures the average scattering of clusters (*Scat*(n_c)) and has already been defined in the *SD* index part.

The *S_Dbw* is calculated using the following equation:

$$S\_Dbw(n_c) = Scat(n_c) + Dens\_bw(n_c) \quad (13)$$

the definition of *S_Dbw* considers both compactness and separation. The number of clusters that minimizes the index is an optimal value.

The Dunn index is calculated using the following equation :

$$D_{n_c} = \min_{i=1,\ldots,n_c} \left\{ \min_{j=i+1,\ldots,n_c} \left[\frac{\frac{1}{|C_i||C_j|} \sum_{x \in C_i, y \in C_j} d(x,y)}{\max_{k=1,\ldots,n_c} \left(\frac{\sum_{x \in C_k} d(x,c_k)}{|C_k|} \right)^2} \right] \right\} \quad (14)$$

where c_i represents the *i*-cluster of a certain partition, d(x,y) is the distance between data points x and y, where x belongs to cluster *i* and y belongs to cluster *j* , d(x, c_k) is the distance of data point x to the cluster centre that it belongs to, $|C_k|$ is the number of data points in cluster K.

The main goal of the measure is to maximize the intercluster distances and minimize the intracluster distances. Therefore, the number of clusters that maximizes D is taken as the optimal number of clusters.

The DB index is calculated using the following equation:

$$DB = \frac{1}{n} \sum_{i=1}^{n} \max_{i \neq j} \left\{ \frac{S_n(Q_i) + S_n(Q_j)}{S(Q_i, Q_j)} \right\} \quad (15)$$

where *n* is the number of clusters, S_n is the average distance of all objects from the cluster to their cluster center, $S(Q_i, Q_j)$ denotes the distance between centres of clusters.

The Davies-Bouldin index is a function of the ratio of the sum of within-cluster scatter to between cluster separation. When it has a small value it exhibits a good clustering.

The following formula is used to calculate the Silhouette index:

$$S(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}} \quad (16)$$

where $a(i)$ is the average dissimilarity of *i*-object to all other objects in the same cluster, Euclidian distance is used to calculate the dissimilarity; and $b(i)$ is the average dissimilarity of *i*-object to all objects in the closest cluster.

The formula indicates that the silhouette value is in the interval [–1, 1]:

- Silhouette value is close to 1: means that the sample is assigned to a very appropriate cluster.
- Silhouette value is about 0: means that that the sample lies equally far away from both clusters, it can be assigned to another closest cluster as well.
- Silhouette value is close to –1: means that the sample is "misclassified".

The partition with the largest overall average silhouette means the best clustering. So, the number of clusters with the maximum overall average silhouette width is taken as the optimal number of clusters. This index is defined as follows:

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \quad (17)$$

where *S* is the sum of distances over all pairs of patterns from the same cluster, *L* is the number of pairs for calculating S_{min} and S_{max}, S_{min} is the sum of the *l* smallest distances if all pairs of patterns are considered, and S_{max} is the sum of the *l* largest distances out of all pairs. It can be seen that a small value of *C* indicates a good clustering.

4 Experiments

We conducted our experiments on Intel® 4, 2.00 GHz CPU, 512 MB RAM running Windows XP Dell PC. The proposed MOKGA approach and the utilized cluster validity algorithms have been implemented using Microsoft Visual Studio 6.0 C++. We used two data sets in the evaluation process. The first data set is the Iris dataset [17]. It contains 150 instances each having 4 attributes; it has three clusters each has 50 instances. The Iris dataset is a famous dataset widely used in pattern recognition and clustering. One cluster is linearly separable from the other two and the latter two are not exactly linearly separable from each other. The second data set is the Ruspini dataset with 75 instances with 2 attributes and integer coordinates: $0 < X < 120$, $0 < Y <$

160, which might be naturally grouped into 4 sets [16]. The Ruspini dataset is popular for illustrating clustering techniques.

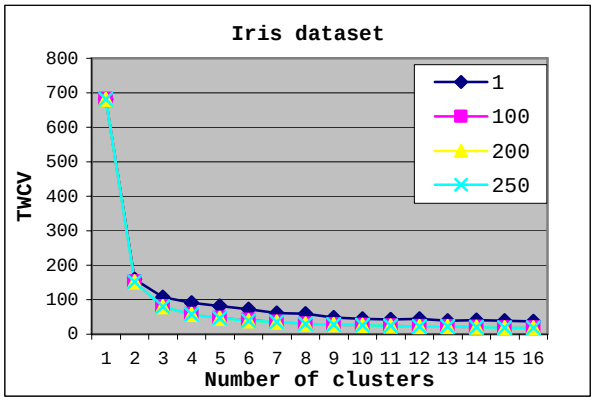

Figure 2: Pareto-fronts for IRIS dataset

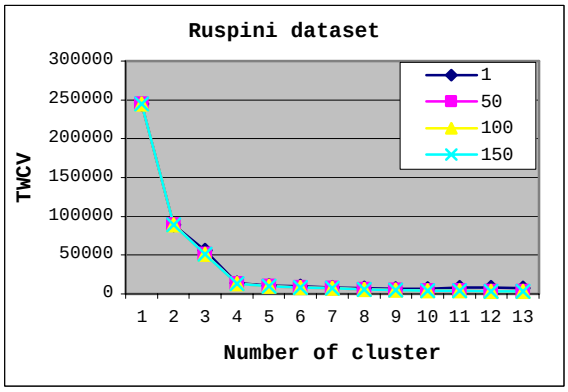

Figure 3: Pareto-fronts for RUSPINI dataset.

Table 1: IRIS DATASET TWCV FOR K=6

Iteration	TWCV
1	72.60164
50	40.4213
100	39.9218
150	39.6762
250	39.5762

Table 2: RUSPINI DATASET TWCV FOR K=12

Iteration	TWCV
1	8331.376
50	3555.116
100	3524.366
150	3513.254

We have run the proposed genetic algorithm based approach ten times with the following parameters: population size=100, t_dom (number of comparison set=10) and crossover= 0.8 and mutation=0.01 and we used 250, and 150 as the maximum number of generations for the Iris and Ruspini datasets, respectively. Finally, we picked the range [2, 20] for finding the optimal number of clusters for both experiments.

After running the algorithm for the Iris and Ruspini datasets, the changes in the pareto-optimal front are displayed in Figure 2, and Figure 3, respectively, for different generations; demonstrating how the system converges to an optimal pareto-optimal front. As the actual change in the value of TWVC is not reflected in the curves in Figure 2 and Figure 3, some key TWVC values are reported in Table 1 and Table 2, respectively.

After we get the pareto optimal front, we tested and analyzed the obtained results for the two data sets using the six indexes: Dunn index, Davies-Bouldin index, Silhouette index, C index, SD index and S_Dbw index. The results are reported in Figures 4-7. Finally, we compared our results with the corresponding results reported in [16, 17].

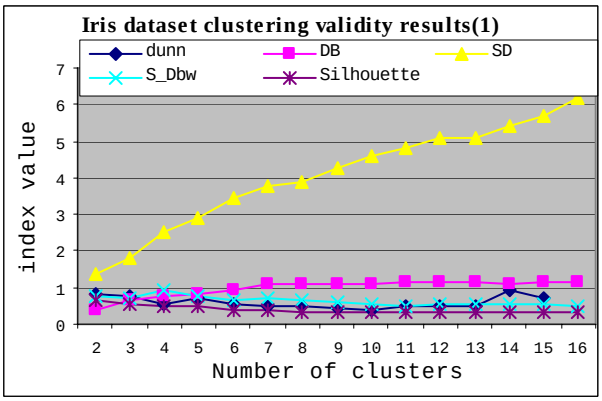

Figure 4: Validity results of five indexes for the IRIS dataset

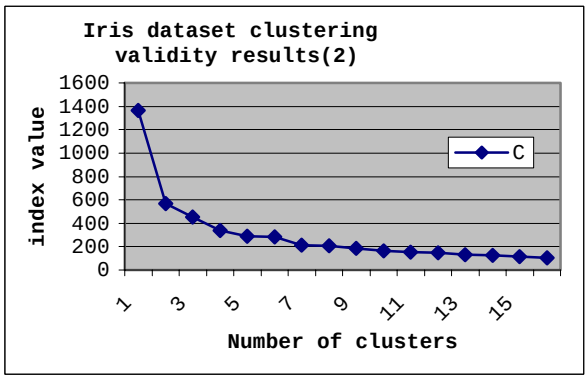

Figure 5: C-index validity results for the IRIS dataset

According to [17], the optimal number of clusters found for the Iris data is 3, which ranked the second for all the indexes except S-Dbw (see Figure 4 and Figure 5).

During the process, pareto-optimal front changes were given in Figure 2 and Figure 3. The reason for getting a stabilized front for both datasets is the k-means operator and the mutation which was also used in [1]. In the work described in [1], they used 517 instances with 19 attributes and their study found the stabilization in 20 generations. However, objective functions do not converge in the first 20 generations; the partitioning error

keeps reducing slightly because of the k-means and mutation. We make sure it converged by having objective functions without a change between the current and the previous one. In our multi-objective genetic algorithm based approach, we got the same outcome as in [1]. In other words, our method stabilizes in the first 50 generations but it does not converge until around 200 generations for Iris; and close to 150 generations for Ruspini. Because of the lack of space, we show the change for one cluster *k* (finally converging ones).

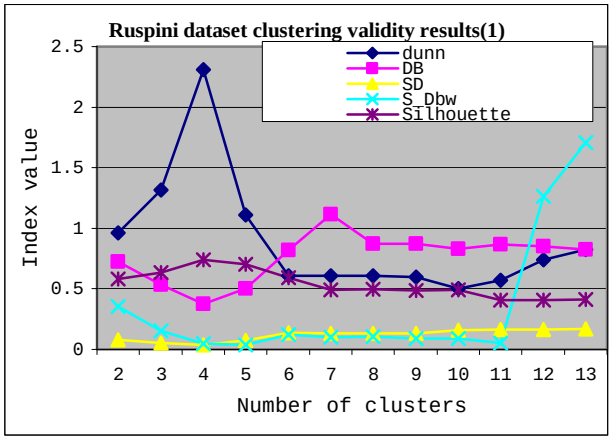

Figure 6: Validity results of five indexes for the RUSPINI dataset

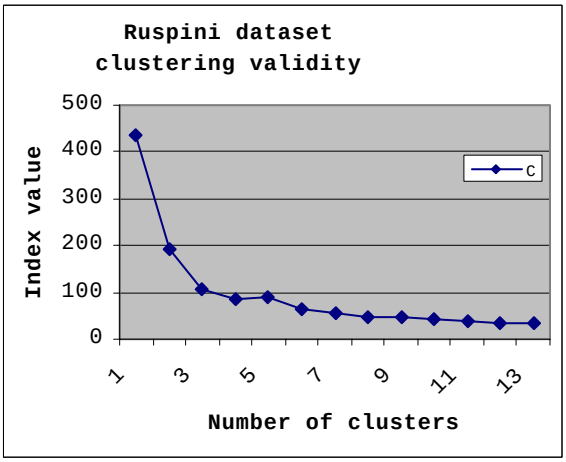

Figure 7: C-index validity results for the RUSPINI dataset

For the Ruspini dataset, it is naturally grouped into 4 clusters as reported in [16]. Not only we have 4 in our pareto optimal front as it can be easily seen from the curves plotted in Figure 6 and Figure 7, but also we got this value as the best in all the cluster validity analysis indexes except C-index. However, 4 can be considered as the best for the C-index as well if it is deemed slight changes after 4 has converged. This finding is consistent with the results found before. Actually, C index is likely to be data dependent and the behavior of the index may change when different data structures were used.

5 Conclusions

In this paper, we proposed a multi-objective genetic algorithm called MOKGA to handle the clustering problem. It is the combination of the niched pareto optimal and fast k-means genetic algorithm. In other words, in MOKGA both crossover and mutation operators are used for the evolutionary process, in addition to the *K*-means operator used to make the evolutionary process faster. For the selection, Niched Pareto tournament selection method is used. Additionally, a multiple Pareto-optimal front layer ranking method is proposed to maintain relative consistence population size in the genetic process. In the experiments, it is also verified that this method can help in leading to the global optimal solution set. In the MOKGA process, the distance (Euclidean distance) between the current generation's Pareto optimal front and the previous generation is calculated and counted compared with the threshold, which can be used to decide when to terminate the genetic process. This way, we overcome the difficulty of determining the weight of each objective function taking part in the fitness. Otherwise, the user would have been expected to do many trials with different weighting of objectives as in traditional genetic algorithms. By using MOKGA, we aim at finding the pareto-optimal front to help the user to see many alternative solutions at once. Then, cluster validity index values are evaluated for each pareto-optimal front value, which is considered the optimal *number of clusters* value.

References

[1] Y. Lu, et al, "FGKA: A Fast Genetic K-means Clustering Algorithm," *Proceedings of ACM Symposium on Applied Computing*, pp.162-163, Nicosia, Cyprus, 2004.

[2] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," *Proceedings of IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation*, Vol.1, pp.82-87, Piscataway, NJ., 1994.

[3] D.L. Davies and D.W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, No.1, pp.224-227, 1979.

[4] J. Dunn, Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, Vol.4, pp.95-104, 1974.

[5] P.J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,"

Journal of Comp App. Math, Vol.20, pp.53-65, 1987.

[6] L. Hubert and J. Schultz, "Quadratic assignment as a general data-analysis strategy," *British Journal of Mathematical and Statistical Psychologie*, Vol.29, pp.190-241, 1976.

[7] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1998.

[8] M. Halkidi, M. Vazirgiannis and I. Batistakis, "Quality scheme assessment in the clustering process," *Proceedings of PKDD*, Lyon, France, 2000.

[9] M. Halkidi, M. Vazirgiannis, Clustering "Validity Assessment: Finding the optimal partitioning of a data set," *Proceedings of IEEE ICDM*, California, Nov. 2001.

[10] Gene Expression Data of the Genomic Resources, University of Stanford (Available at: http://genome-www.stanford.edu/ serum/data.html), accessed June 2004.

[11] J. Grabmeier, et al, "Techniques of Cluster Algorithms in Data Mining," *Data Mining and Knowledge Discovery*, Vol.6, pp.303–360, 2003.

[12] A. K. Jain, et al, "Data Clustering: A Review," *ACM Surveys*, Vol.31, No.3, 1999.

[13] K. Tamura, et al, "Necessary and Sufficient Conditions for Local and Global Non-Dominated Solutions in Decision Problems with Multi-objectives," *Journal of Optimization Theory and Applications*, Vol.27, 509-523, 1979.

[14] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," *Doctoral thesis ETH NO. 13398, Zurich: Swiss Federal Institute of Technology (ETH)*, Aachen, Germany: Shaker Verlag, 1999.

[15] K. Deb et al., "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II," *Proceedings of the Parallel Problem Solving from Nature. Springer Lecture Notes in Computer Science No. 1917*, Paris, France, 2000.

[16] L. Kaufman and P.L. Rouseeauw, *Finding group in data: An introduction to cluster analysis*, John Wiley & Sons. New York, p.100, 1990.

[17] H.P. Friedman and J. Rubin, "On some invariant criteria for grouping data," *Journal of the American Statistical Association*, 62:1159-1178, 1967.

[18] Microarray Data Analysis: Direct Gene Sample Correlations, Gene Network Science, Inc. (c). 2001.

[19] M. Neef, D. Thierens, & H. Arciszewski, A Case Study of a Multi-objective Elitist Recombinative Genetic Algorithm with Coevolutionary Sharing. In Angeline, P. (Ed.), *Proc. of the International Congress on Evolutionary Computation*, pp.796-803. Priscatawy. 1999.

[20] W. Shannon, R. Culverhouse J. Duncan. Analyzing microarray data using cluster analysis, *Pharmacogenomics*, Vol.4, No.1, pp.41-52, 2003.

[21] T. Kohonen, Self-organizing Maps, Springer-Verlag, 1997.

[22] P.J. Waddell, H. Kishino, Cluster Inference Methods and Graphical Models Evaluated on NCI60 Microarray Gene Expression Data, *Genome Informatics*, Vol.11, pp.129-140, 2000.