Knowledge-Based Systems 56 (2014) 108–122

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Reporting and analyzing alternative clustering solutions by employing multi-objective genetic algorithm and conducting experiments on cancer data [☆]

Peter Peng [a], Omer Addam [a], Mohamad Elzohbi [a], Sibel T. Özyer [b], Ahmad Elhajj [c], Shang Gao [a], Yimin Liu [a], Tansel Özyer [d], Mehmet Kaya [e], Mick Ridley [c], Jon Rokne [a], Reda Alhajj [a,f,*]

[a] *Department of Computer Science, University of Calgary, Calgary, Alberta, Canada*
[b] *Department of Computer Engineering, Cankaya University, Ankara, Turkey*
[c] *Department of Computing, University of Bradford, Bradford, UK*
[d] *Department of Computer Engineering, TOBB University, Ankara, Turkey*
[e] *Department of Computer Engineering, Firat University 23119, Elazig, Turkey*
[f] *Department of Computer Science, Global University, Beirut, Lebanon*

ARTICLE INFO

Article history:
Received 16 April 2013
Received in revised form 22 September 2013
Accepted 1 November 2013
Available online 14 November 2013

Keywords:
Clustering
Genetic algorithm
Gene expression data
Multi-objective optimization
Cluster validity analysis

ABSTRACT

Clustering is an essential research problem which has received considerable attention in the research community for decades. It is a challenge because there is no unique solution that fits all problems and satisfies all applications. We target to get the most appropriate clustering solution for a given application domain. In other words, clustering algorithms in general need prior specification of the number of clusters, and this is hard even for domain experts to estimate especially in a dynamic environment where the data changes and/or become available incrementally. In this paper, we described and analyze the effectiveness of a robust clustering algorithm which integrates multi-objective genetic algorithm into a framework capable of producing alternative clustering solutions; it is called Multi-objective K-Means Genetic Algorithm (MOKGA). We investigate its application for clustering a variety of datasets, including microarray gene expression data. The reported results are promising. Though we concentrate on gene expression and mostly cancer data, the proposed approach is general enough and works equally to cluster other datasets as demonstrated by the two datasets Iris and Ruspini. After running MOKGA, a pareto-optimal front is obtained, and gives the optimal number of clusters as a solution set. The achieved clustering results are then analyzed and validated under several cluster validity techniques proposed in the literature. As a result, the optimal clusters are ranked for each validity index. We apply majority voting to decide on the most appropriate set of validity indexes applicable to every tested dataset. The proposed clustering approach is tested by conducting experiments using seven well cited benchmark data sets. The obtained results are compared with those reported in the literature to demonstrate the applicability and effectiveness of the proposed approach.

1. Introduction

A multi-objective genetic algorithm based clustering method is described in this paper. Its applicability and effectiveness are demonstrated by using some benchmark datasets, mainly related to gene expression data analysis which constitutes a vital research area of social and scientific impacts. Fortunately, clustering is one of the key methods that can be employed to the benefit of the computational biology and bioinformatics research communities. It allows researchers to identify molecules that demonstrate similar behavior or characteristics and hence could lead to utilizing in the analysis reduced set of molecules by considering representatives from each cluster instead of the whole original set of molecules.

In general, existing clustering techniques require pre-specification of the number of clusters or some parameters that indirectly lead to the number of clusters; and these are is not an easy to predict *a prior* even for experts. Thus, the problem handled in this paper may be articulated as follows: Given a set of data instances (we mainly concentrate on gene expression data), it is required to develop an approach that produces different alternative solutions, and then rank the resulting solutions by conducting validity analysis. In fact, there are always some trade-offs between

[☆] This paper is part of the project sponsored by Scientific and Technical Research Council of Turkey (Tübitak EEEAG 109E241). Tansel Ozyer would like to thank TUBITAK for their support.

[*] Corresponding author.
E-mail address: alhajj@cpsc.ucalgary.ca (R. Alhajj).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.11.003

the quality of a clustering result and the number of clusters. One solution is to view the two elements as two objectives that affect clustering results, i.e., this is naturally a multi-objective optimization problem. The solution of a multi-objective optimization problem is a set of alternatives which in one way can be seen as a Pareto-optimal set or non-dominated set [52].

In general, traditional algorithms for clustering microarray data do not produce the Pareto optimal set, and they do not lead to the optimal number of clusters in the database that they work on. For example, the hierarchical clustering method can get the heuristic overview of a whole dataset, but it cannot relocate objects that may have been 'incorrectly' grouped at an early stage. It can neither tell the optimal number of clusters nor give the non-dominated set. Partitional clustering like *K*-means needs the number of clusters as a predefined parameter, and it may lead to local optimal solutions because it concentrates on a local search from a random initial partitioning. SOM has the same disadvantage in that it requires the number of clusters as *a prior*. Clearly, a more advanced and comprehensive clustering algorithm is needed to get the global pareto-optimal solution set required to give users the best overview of the whole dataset according to the number of clusters and their quality. Further, it is required to get clustering results with the optimal number of clusters.

Clustering different samples based on gene expression is one of the key issues in problems like class discovery, normal and tumor tissue classification, and drug treatment evaluation [1,69]. Scherf et al. [58] applied microarray analysis on the gene expression database for the molecular pharmacology of cancer. It contains 728 genes, 60 cell lines, and 15 cell line groups. Golub et al. [17] applied SOM clustering algorithm on gene expression data containing 38 acute leukemia samples and 50 genes after filtered the whole dataset. SOM automatically grouped the 38 samples into two classes with acute myeloid leukemia (ALL) and acute lymphoblastic leukemia (AML). They further used SOM to group the samples into four classes. Subclasses of ALL, namely, B-lineage ALL and T-lineage ALL were distinguished [17]. It has been indicated that clustering samples can be used to identify fundamental subtypes of any cancer [58].

Clustering analysis can also be used to find direct gene-sample correlations. BiCluster [13] enables Gene/Condition correlation analysis that can lead to molecular classification of disease states, identification of co-fluctuation of functionally related genes, functional groupings of genes, and logical descriptions of gene regulation, among others. It is a starting point for understanding the large-scale network [13,44]. Domany [15] proposed a Coupled Two-Way Clustering (CTWC), which breaks down the total dataset into subsets of genes and samples that can reveal significant partitions into clusters. It provides clues about the function of genes and their roles in various pathologies.

The main contribution of this paper is a comprehensive and general purpose clustering approach that considers multiple objectives in the process and its application for clustering microarray data. The proposed approach has two components:

1. Multi-objective *K*-means Genetic Algorithm (MOKGA) based clustering approach has been developed to deliver a Pareto optimal clustering solution set without taking weight values into account. Otherwise, users need to consider several trials weighting with different values until a satisfactory result is obtained.
2. Cluster validity analysis and voting technique have been employed to evaluate the obtained candidate optimal number of clusters, by applying some of the well-known cluster validity techniques, namely Silhouette, C index, Dunn's index, DB index, SD index and S-Dbw index, to the clustering results obtained from MOKGA. It gives one or more options for the optimal number of clusters.

The applicability and effectiveness of the described clustering approach and clustering validity analysis process are demonstrated by conducting experiments using seven datasets from various domains: two breast cancer datasets, namely GSE12093 and GSE9195, Fig2data, NCI60 cancer data, Leukemia data sets available at Genomics Department of Stanford University, UCI machine learning repository, Iris at Genome Research MIT and Ruspini dataset.

The balance of the paper is organized as follows. Section 2 is an overview of the clustering approaches used primarily in the microarray data analysis area. Section 3 is devoted to the development of the new clustering system MOKGA for clustering both gene expression and general datasets. Section 4 reports experimental results on five datasets to test the applicability, performance, and efficiency of the system. Section 5 discusses the advantages and disadvantages of the proposed approach in comparison with other existing methods; conclusions are made and future research directions are suggested.

2. Related work

Existing clustering techniques which have been used for gene expression data can be classified into hierarchical clustering [28,48], partitioning [33], graph-based [44] and model-based [61,67] approaches.

Hierarchical clustering algorithms have been widely used in the area of gene expression data analysis. For example, Waddell and Kishino [67] applied hierarchical clustering based on partial correlations on NC160 gene expression data to find a tight and closed set of genes, and the interaction of important genes of the cell cycle. A tree structure *dendogram* is used to illustrate the hierarchical clustering [20,28,48]. Hierarchical clustering methods suffer from different aspects as stated by statisticians, including robustness, non-uniqueness, and inverse interpretation of the hierarchy [45,63]. Segal and Koller [59] proposed probabilistic abstraction hierarchies (PAH). This method improved the performance of traditional hierarchical clustering by handling the drawbacks mentioned above.

K-Means is a commonly used algorithm for partition clustering [33]. The purpose of *K*-Means clustering is the optimization of an objective function that is described by the equation:

$$E = \sum_{i=1}^{c} \sum_{x \in C_i} d(x, m_i) \tag{2.1}$$

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.

Self Organizing Maps (SOM) [30] is popular in vector quantization. It uses an incremental approach; points (patterns) are processed one-by-one. The shortcoming of SOM is that the size of the two dimensional grid and the number of nodes have to be predetermined. It suits well when prior information about data distribution is not available. Double self organizing maps (DSOM) technique [68] is also used for gene expression data clustering. In DSOM, each node does not have only an *n*-dimensional synaptic weight vector, but also a 2-dimensional position vector.

The model-based approach [53] is a promising technique, which assumes that data are generated by a mixture of finite number of probability distributions. In this approach, each cluster represents a probability distribution and a likelihood-based framework can be used. The Bayesian method is a model-based approach used in gene expression data analysis. Barash et al. [2,3] applied the Bayesian method on gene-expression time series data to study the response of human fibroblasts to serum. Gaussian mixture model is used in the method. They found the dynamic nature of gene expression time series during clustering. Mar and McLachlan [39] proposed a mixture model-based algorithm (EMMIX-GENE) for the clustering of tissue samples and presented a case study involving the application of EMMIX-GENE to breast cancer data.

Graph-based clustering methods translate a clustering problem into a graph partitioning problem by creating a weighted similarity graph and linking each gene to other genes that are more than some threshold similar to it [4]. The study by Ben et al. [4] tries to make cliques for the clustering purpose. Examples of this approach are the Two-Way Clustering Binary tree [13] and the Coupled Two-Way Clustering [53].

After data clustering and data partitioning into subgroups, the validity of the result must be checked [46]. Levine introduced a cluster validation method based on resampling [34]. Roth [54] tested the stability by clustering two sets of equal size data sampled from $2n$ size source data and calculated the rates that the algorithm clusters the same object into different clusters. A slight modification of the noise may then alter the cluster structure significantly. The disadvantage of this method is that it is unsuitable for very sparse data. In this case, dilution can eliminate some of the underlying models [6,34].

Bootstrapping cluster analysis begins by creating a number of simulated datasets based on statistical models, such as the analysis of variance (ANOVA) model [31]. Other widely accepted criteria used by clustering algorithms are compactness of the clusters and their separateness. These criteria should be validated and optimal clusters should be found such that the correct input parameters must be given to the satisfaction of optimal clusters. Some clustering validity techniques used for the validation task include Dunn index [64], Davies–Bouldin (DB) index [8], Silhouette index [25], C index [66], SD index [43] and S_Dbw index [17], among others. Dunn's index uses the dispersion parameter, which is prone to noise since it uses the maximum of pairwise distance of objects in the same cluster as a parameter. Davies–Bouldin (DB) uses the ratio of scattering (uses Euclidean distance to calculate the scattering ratio) of objects within a cluster and the scattering of cluster centers. It considers the average case by using the average error of each class.

C-index uses the within cluster pairwise dissimilarity. Further, according to the number of pairs in the within cluster pairs, minimum and maximum summation of the number of pairwise object distance parameters are used in the calculation. However, this method is not recommended since it is likely to be data dependent [7]. Examples of other cluster validity approaches used in gene expression data analysis include Principal Component Analysis (PCA) [5] and Gap statistic [65]. PCA is a statistical method that can improve the extraction of cluster structure and compare clustering solutions [5]. Gap statistic utilizes within-cluster distance to determine the "appropriate" number of clusters in a dataset. It is good at identifying well-separated clusters, but it does not produce satisfactory results for not-well-separated data and data concentrated on a subspace [21].

Our proposed Multi-object GA based clustering algorithm has the salient randomization feature that originates from the classical k-means algorithm where random sampling of object is needed at the start of the clustering process and quality metric converges iteratively. A randomized clustering is essentially a stochastic process, i.e., clustering data objects or observations with the belief that events occur in random orders [19,41]. Even properties of data are unknown; the assumption that it follows certain stochastic behavior typically suffices to achieve unsupervised learning goals. In the machine learning direction, where objects or observations are taken globally without distribution estimation, randomization typically means a good sampling process stemmed from prior knowledge of data [26]. Convergence in randomized clustering process is crucial because it specifies termination condition of the process [60] and can be useful in genetic algorithm based clustering methods [42,47]. In data distribution direction, observations are gauged to fit certain probabilistic distribution such as Gaussian or Mixed Gaussian, and clustering process is statistical manipulations on distributions [62,23,9].

The method described in this paper assumes that a clustering process may have several objectives by nature. So, it is difficult to find the optimal solution to the satisfaction of all the objectives. Rather than using a fixed threshold value and/or a *prior* specified fixed number of clusters, this paper is keen on giving a range for the number of clusters parameter and finding a set constituting pareto optimal solution to find the superior results in the sense that there is no other point which can be superior to the pareto-optimal solution. This idea differs from traditional multi-objective algorithms that scalarize the objectives by assigning subjective weights to each function, e.g., [11,14,16,24,35,40,57]. Hence, we do not need to consider weights in the system. In addition, using a genetic algorithm with recombination and mutation, we can find the global optimum solution using appropriate system parameters. We have already demonstrated the benefit of the methodology described in this paper to some interesting applications like data partitioning for skyline computation [51] and fuzzy association rules mining [29,32]. Finally, to allow for scalability, we have utilized the divide and conquer concept to partition the data into subsets where each subset is manageable by a single traditional machine [49,50]. Then, the final solution is achieved by combining the partial solutions in a hierarchical way where after clustering the subsets individually, we concenrtate on clustering the centroids in order to incrementally combine the solutions.

In summary, the method presented and analyzed in this paper is unique in presenting the set of solutions in the pareto optimal front and analyzing their validity to select the most approporiate from all valid candidate solutions. The comparison of the results of validity analysis with the known single results reported in the literature for each considered data set supports the applicability and effectiveness of the approach described in this paper.

3. Description of the proposed approach

A clustering approach named Multi-Objective Genetic *K*-means algorithm (MOKGA) is described here. It is a general-purpose approach for clustering datasets from various domains as demonstrated by the test results reported in Section 4. It has been developed on the basis of the Fast Genetic K-means Algorithm (FGKA) [38] and the Niched Pareto Genetic Algorithm [22].

After running the multi-objective *K*-means genetic algorithm, the Pareto-optimal front giving the optimal number of clusters as a solution set can be obtained. The system then analyzes the clustering results found with respect to various cluster validity techniques proposed in the literature, namely Silhoutte, *C* index, Dunn's index, SD index, DB index, and S_Dbw index. These techniques have been chosen arbitrarily. Other techniques may be used without affecting the overall outcome because the target is achieved by applying majority voting.

This section is organized as follows. The objectives of the Multi-Objective Genetic *K*-means algorithm (MOKGA) are discussed in Section 3.1. The chromosome representation process in MOKGA is introduced in Section 3.2. Section 3.3 presents the fitness evaluation and selection. Section 3.4 discusses the mutation and cross-over operations. Implementation details are described in Section 3.5.

3.1. The utilized objectives

During the clustering process three objective functions are defined, namely maximizing homogeneity and separateness and minimizing the number of clusters. These objectives do conflict as the number of clusters decreases, the values of the other two objectives will be negatively affected. In other words, the first two objectives are defined as: minimizing the partitioning error and minimizing the number of clusters. To partition the N objects into K clusters, one goal is to minimize the Total Within-Cluster Variation (*TWCV*) and maximize the separateness of the clusters. The value of TWCV is computed as:

$$TWCV = \sum_{n=1}^{N}\sum_{d=1}^{D} X_{nd}^2 - \sum_{k=1}^{K}\frac{1}{Z_k}\sum_{d=1}^{D} SF_{kd}^2 \tag{3.1}$$

where $X_1, X_2, \ldots, X_N$ are the N objects, X_{nd} denotes feature d of object X_n (n = 1 to N), Z_k denotes the number of objects in cluster k, and SF_{kd} is the sum of the dth features of all the objects in cluster k:

$$SF_{kd} = \sum_{\overrightarrow{X_n}\in G_k} X_{nd}, \quad (d = 1, 2, \ldots D). \tag{3.2}$$

Separateness of clusters is measured using one of the following four equations:

$$\textit{Centroid Linkage}: d(C,D) = d(v_C, v_D) \tag{3.3}$$

$$\textit{Complete Linkage}: d(C,D) = \max_{x\in C, y\in D} d(x,y) \tag{3.4}$$

$$\textit{Average Linkage}: d(C,D) = \frac{1}{|D||C|}\sum_{x\in C, y\in D} d(x,y) \tag{3.5}$$

$$\textit{Average ToCentroid Linkage}: d(C,D) = \frac{1}{|D|+|C|}\left[\sum_{x\in C} d(x, v_D) + \sum_{y\in D} d(y, v_C)\right] \tag{3.6}$$

The other objective function minimizes the *number of clusters* parameter.

$$F = \min(\text{number of clusters}) \tag{3.7}$$

After running the algorithm, the aim is obtaining the first Pareto optimal front having the best partitioning with the least number of clusters as an optimal solution set.

3.2. Chromosome encoding

The coding of the individual population is a chromosome of length N. Each gene in the chromosome takes a value from the set of k clusters $\{1, 2, \ldots, k\}$ and represents an object. The value indicates the cluster to which the corresponding object belongs. 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 a random value from the interval $[1, k]$. The process is repeated P times to produce P chromosomes which form the initial solution set on which the various genetic operators will be applied, as described in the sequel, leading to the final solution set. The value of P is arbitrarily determined.

3.3. Fitness evaluation and selection

The fitness value for each chromosome is computed based on the average *TWCV* of the clusters in the solution represented by the chromosome and on the separateness of these clusters. In this paper, the Niched Pareto tournament selection scheme is used for the selection process in the multi-objective genetic clustering system. The scheme is described as follows: Two candidates for selection are picked randomly from the population, and then each of the candidates is compared against each individual in the comparison set, which is the set from the previous result set, then the set with the new candidate is compared with the previously selected set. If the candidate is dominated by the comparison set, it will be deleted from the population. In this system, if both candidates are non-dominated, they will be kept in the population. This is different from the original Niched Pareto Tournament Selection where if neither of the two is dominated by the comparison set then they will use sharing to choose a winner [22], which is not necessarily in this system.

3.4. Crossover and mutation

Some initial experiments demonstrated that one-point crossover produces better fitness values than multi-point attempts. So, in this study one-point crossover operator is applied on two randomly chosen chromosomes. The crossover operation is carried out on the population with crossover rate p_c. After the crossover, assigned cluster numbers for each gene are renumbered beginning from a_1 to a_N. For example, give two chromosomes having 3 and 5 clusters, respectively:

$$\begin{aligned}&\text{Number of clusters} = 3: \quad 1\ 2\ 3\ 3\ 3;\\ &\text{Number of clusters} = 5: \quad 1\ 4\ 3\ 2\ 5,\end{aligned}$$

and assume they need to have a crossover at the third location, we will get 1 2 3 2 5 and 1 4 3 3 3, which are then renumbered to get the new number of clusters parameters:

$$\begin{aligned}&\text{Number of clusters} = 4: \quad 1\ 2\ 3\ 2\ 4 \quad (\text{for } 1\ 2\ 3\ 2\ 5);\\ &\text{Number of clusters} = 3: \quad 1\ 2\ 3\ 3\ 3 \quad (\text{for } 1\ 4\ 3\ 3\ 3)\end{aligned}$$

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

$$P_i = \frac{1.5^* d_{\max}(\overrightarrow{X_n}) - d(\overrightarrow{X_n}, \overrightarrow{C_k})}{\sum_{k=1}^{K}(1.5^* d_{\max}(\overrightarrow{X_n}) - d(\overrightarrow{X_n}, \overrightarrow{C_k}))} \tag{3.8}$$

where $i = (1, 2, \ldots, k)$ and $d(X_{n,} C_k)$ denotes the Euclidean distance between object X_n and the centroid C_k of the kth cluster, $d_{\max}(X_n) = \max_k\{d(X_{n,} C_k)\}$, and the constant 1.5 has been arbitrarily chosen to guarantee that the computed probability value is greater than zero for every gene i, which is necessary for the convergence to be achieved; having the mentioned constant greater than 1 will guarantee this. Here, p_i represents the probability interval of a mutating gene assigned to cluster i (e.g., Roulette Wheel). Using this method, the probability of changing gene value a_n to a cluster number k is greater if X_n is closer to the centroid of the kth cluster G_k.

3.5. Implementation details

The clustering system described in this paper consists of two components: the Multi-Objective Genetic K-means Algorithm (MOKGA) cluster and the cluster validity component. The implementation details are described next.

MOKGA uses a list of parameters to drive the evaluation procedure as in other genetic types of algorithms: including population size (the number of chromosomes), t_dom (the number of comparison set) representing the assumed non-dominated set, crossover, mutation probability, and the number of iterations for the execution of the algorithm to obtain the result. Subgoals 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, it is expected that the system can 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.

Initially, the *current generation* is assigned to zero. Each chromosome takes the *number of clusters* parameter within the range 1 to the maximum number of clusters given by the user. A population with the specified number of chromosomes is created randomly by using the method described by Rousseeuw [55], where data points are randomly assigned to each cluster at the beginning and the rest of the points are randomly assigned to clusters. By using this method, we can avoid the generation of illegal strings, which means some clusters do not have any pattern in the string.

Using the current population, the next population is generated and the 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 the k-means operator [38] are then applied to reorganize each object's assigned cluster number. Finally, we will have twice the number of individuals after the Pareto domination tournament. The ranking mechanism used by Zitzler [72] is applied to satisfy the elitism and diversity preservation. This halves the number of individuals in the population to be moved to the next iteration.

The first step in the construction of the next generation is the selection using Pareto domination tournament. 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 chromosome in the population. In the selection part, t_{dom} individuals are randomly picked from the population. Two chromosome candidates are randomly selected from the current population except those in the comparison set (*population size*-t_{dom}), and 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 clusters value than all of the chromosomes in the comparison set, then it is dominated by the comparison set and will be deleted from the population permanently. Otherwise, it resides in the population. The corresponding pseudo code is given below:

```
Function selection
Begin
      shuffle(random_pop_index, number_of_rules) /*Re-randomize random index array*/
   candidate_1=random_pop_index[0]
   candidate_2=random_pop_index[1]
   candidate_1_dominated = false;
   candidate_2_dominated = false;
   For comparison_set_index = 3 to $t_{dom}$ + 3 do /* Select $t_{dom}$ individuals randomly from current population S*/
   comparison_individual = random_pop_index [comparison_set_index]
   If S[comparison_individual] dominates S[candidate_1] then candidate_1_dominated = true
   If S[comparison_individual] dominates S[candidate_2] then candidate_2_dominated = true
  End For
   If (candidate_1_dominated AND candidate_2_dominated) delete_rule(candidate_1, candidate_2);
   If (candidate_1_dominated AND not candidate_2_dominated) delete_one_rule(candidate_1);
  If (not candidate_1_dominated AND candidate_2_dominated) delete_one_rule(candidate_2);
End selection
```

After the Pareto domination tournament, the dominated chromosome is deleted from the population. The next step is the crossover process. One point crossover is used in the employed multi-objective genetic clustering approach. An index into the chromosome is selected and all data beyond that point in the chromosome are swapped between the two parent chromosomes. The resulting chromosomes are the children. The pseudo code of the function that performs the crossover process is given next:

```
Function crossover
Begin   /* Randomly chose the two chromosomes*/
     Chromosome_1 = rand()% biggest chromosome index
     Chromosome_2 = rand()% biggest chromosome index
     /*Randomly choose the cross point*/
     cross_point = rand()% length of the chromosome
     Swap (Chromosome_1, Chromosome_2, cross_point)
End crossover
```

Mutation is applied to the population in the next step by randomly changing the values in the chromosome according to some probability distribution, as discussed in Section 3.4. The pseudo code of the mutation function is given next:

```
Function mutation
Input: population $P\ (S_1, S_2, \ldots, S_J)$, Mutation probability MP
Output: population $P'(S'_1, S'_2, \ldots, S'_J)$
Begin
    For j = 0 to J do /* for each solution $S_J$ in population P*/
    SD=0; /*summation of distribution*/
      $\overrightarrow{c_1} \ldots \overrightarrow{c_k}$ = CalCentroids($S_J$) /* calculate the centre point for each cluster*/
   For n=1 to N do /*for each data point in $S_J$ */
      If rand() < MP then
        d_max = 0.00;
        For k = 1 to K /* for each cluster */
            $d_k$ = calEuclideanDistance($\overrightarrow{X_n}, \overrightarrow{c_k}$) /* distance from data to cluster centre*/
            d_max = max(d_max, $d_k$)
          $SD = SD + (1.5 \times d\_\max(\overrightarrow{X_n}) - d(\overrightarrow{X_n}, \overrightarrow{c_k}))$
        End For
        $p_1$ = (1.5 $*$ d_max-$d_1$)/SD /* Mutation probability for cluster 1*/
        For k = 2 to K
            $p_k$ = (1.5 × d_max − $d_k$)/SD + $p_{k-1}$; /* Mutation probability for cluster 2~ CLUSTER*/
        End for
        $S'_j.a'_n$ = a cluster number, randomly chose according to the distribution $p_{1,}\ p_{2,} \ldots\ p_k$
      End if MP
   End for n
  End for j
End mutation
```

The *K*-means operator is applied last to reanalyze each chromosome gene's assigned cluster value. It calculates the centroid for each cluster and re-assigns each gene to the closest cluster. In other words, applying K-means helps in quickly rectifying any unwanted outcome from the crossover operator; it is like a confirmation step to guarantee each object belongs to its cluster. Hence, the *K*-means operator is used to speed up the convergence process by replacing a_n by a'_n, for $n = 1$ to N simultaneously, where a'_n is the closest to object X_n in Euclidean distance. The pseudo code for the K-means operator is:

```
Function K-Means operator
Input: population $P\,(S_1, S_2, \ldots, S_J)$
Output: population $P'(S'_1, S'_2, \ldots, S'_J)$
Begin
  For j = 1 to J do /* each solution in a population P*/
    $\vec{c_1} \ldots \vec{c_k}$ = CalCentroids($S_J$) /* calculate the centre point for each cluster*/
      For n = 1 to N do /* each data point in a solution*/
        $d_{min}$ = MAX_NUMBER
        For k = 1 to K do /* K is maximum cluster number*/
          /* calculate the Euclidean distance from the data point to each cluster centre*/
          $d_k = \text{calEuclideanDistance}(\vec{X_n}, \vec{c_k})$
        If ($d_k < d_{min}$) then /* a closer centroid is found*/
          $d_{min} = d_k$;
          $k_{min}$ = k;
        End If
      End For
          $S'_j.a'_n = k_{min}$ /* assign the closet cluster number to the data point*/
      End For
  End For
End K-means operator
```

After all the operators have been applied, twice the number of individuals is produced. After having the Pareto dominated tournament, we cannot give an exact number equal to the initial population size because at each generation randomly picked candidates are selected for the survival test leading to the deletion of one or both, in case dominated. To half the number of individuals, the ranking mechanism proposed by Zitzler [72] is employed. Thus, the individuals obtained after crossover, mutation, and the K-means operator are ranked, the best individuals are picked to place in the population for the next generation.

The approach picks the first l individuals by considering the elitism and diversity among $2l$ individuals. Pareto fronts are ranked. Basically, we find the Pareto-optimal front and remove individuals of the Pareto-optimal front from the $2l$ set and place them in the population to run in the next generation. In the remaining sets we get individuals constituting the first Pareto-optimal front and put them 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 dimensions into one. We then sum the normalized values of the objective functions for each individual. These are sorted in increasing order and each individual's total difference from its individual pairs is calculated. Individuals are placed in the population based on decreasing differences, and then we keep placing from the top as many individuals as we need to complete the number of individuals in the 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. This method was also suggested as a solution for the elitism and diversity for improvement in NSGA-II. For example, in order to get 20 chromosomes from the population, we select 10 chromosomes from the Pareto front, delete them from the current population, then get 8 chromosomes from the Pareto front in the current population, delete them from the population. Suppose we have 6 chromosomes in the current population, we take 2 chromosomes that have the largest distance to their neighbors using the ranking method mentioned above. Finally, if the maximum number of generations is reached, or the Pareto front remains stable for 50 generations, then the process is terminated; otherwise we proceed to determine the next generation.

4. Experimental results

To evaluate the performance and efficiency of the proposed system consisting of the MOKGA clustering approach and cluster validity analysis, experiments were conducted using a personal computer running Windows 7. The MOKGA clustering approach was implemented using MS Visual C++.

Both widely used general datasets and microarray datasets have been used to test the proposed framework. This demonstrates that the framework described in this paper works not only for microarray (gene expression) data but also for general clustering as well. For example, the two datasets Iris and Ruspini that are widely used in testing clustering approaches described in the literature have been used to test the general MOKGA approach [43,44].

Five gene expression datasets, Fig2data, cancer (NCI60), Leukemia and two breast cancer datasets were used to test the performance and accuracy of the system for gene expression data. Among them, Fig2data data is used for clustering genes, while cancer (NCI60) and Leukemia data sets are used for group cell samples. The description and testing results of the five datasets are discussed in the following sections.

The aforementioned different cluster validity indexes have been used to validate the result. Minimal SD index indicates an optimal cluster number, while maximal Dunn index shows the optimal number of clusters as it maximizes intercluster distances and minimizes the intracluster distances. The DB index is a function of the ratio of the sum of within-cluster scattering to between clusters separation, a small value exhibits a good clustering. Silhouette value is in the interval $[-1, 1]$; a value close to 1 means the sample has been assigned to a very appropriate cluster, and 0 means the sample lies equally far away from both clusters, while close to -1 means the sample is misclassified.

4.1. The Ruspini dataset

The Ruspini dataset [56] is popular for illustrating clustering techniques. It has 75 instances with 2 attributes and integer coordinates: $0 < X < 120$, $0 < Y < 160$, which might be naturally grouped into 4 sets.

In one study [56], four clusters were reported as the best clustering solution for the Ruspini dataset using numerical methods. In another independent study, Cole tested the Ruspini dataset using general genetic algorithms [12]. The same number of clusters was obtained using genetic algorithms by Calinski and Harabasz criterion, Davies and Bouldin cluster validity methods. Values of the major parameters used for the genetic algorithm in this study are: number of iterations = 100, range of exponential mutation rate: from 10.0 to 0.000001, population size = 200, and crossover probability = 1.00.

The multi-objective genetic algorithm-based approach proposed in this paper was run ten times with the following parameters: population size = 100, *t_dom* (the number of comparison set = 10), crossover = 0.8 and mutation = 0.01. Threshold = 0.1 has been used to check if the population stops evolution for 50 generations and if the process needs to be stopped. The range of [1,10] was picked for finding the optimal number of clusters.

Changes in the Pareto-optimal front by running the algorithm for the Ruspini dataset are displayed in Fig. 4.1. It demonstrates

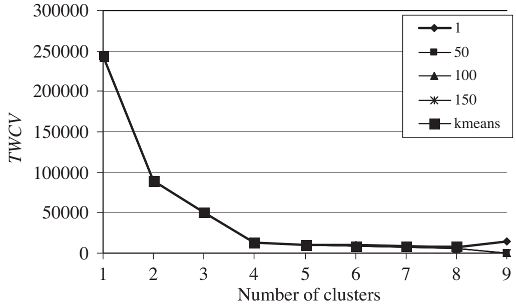

Fig. 4.1. Pareto-fronts for Ruspini dataset.

Table 4.1
Ruspini dataset *TWCV* for $k = 8$.

Iteration	TWCV
1	7718.25
50	6158.25
100	6157.50
150	6149.63
k-means	8185.5

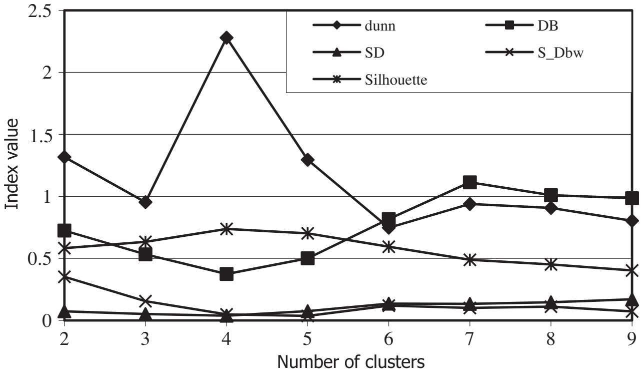

Fig. 4.2. Ruspini dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices.

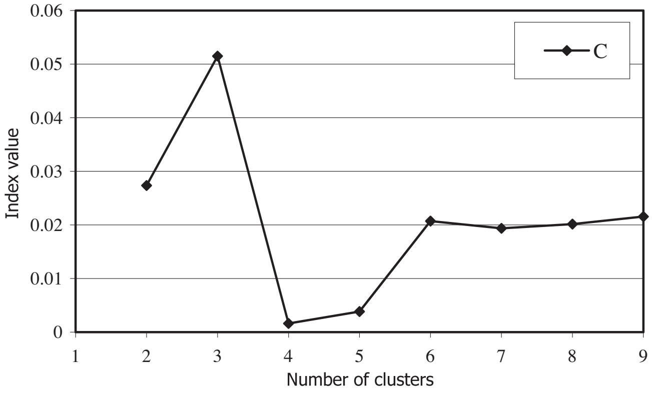

Fig. 4.3. Ruspini dataset cluster validity results using C index.

how the system converges to an optimal Pareto-optimal front. Key *TWVC* values are reported in Table 4.1 because the actual change in the value of *TWVC* is not reflected in Fig. 4.1 where the values are very close and all the five curves almost overlap due to the scale used.

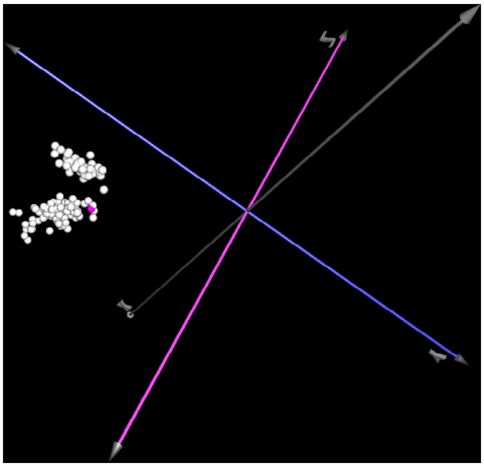

Fig. 4.4. The real cluster distribution visualized with the labels from the original Iris dataset: Iris dataset clustering results from [8].

We run the six cluster validity indexes on the Ruspini dataset. The test results are reported in Figs. 4.2 and 4.3 for five indexes and for the C-index, respectively; we separated the C-index because it works under a different scale. From the curves plotted in the two figures, not only 4 is in our Pareto optimal front, also this value is the best for all the cluster validity analysis indexes. This finding is consistent with the results obtained before and reported by other researchers [12,56].

4.2. The Iris dataset

The Iris dataset is a famous dataset widely used in pattern recognition and clustering. It is a 4-attributes dataset containing 150 instances; it has three clusters each has 50 instances. One cluster is linearly separable from the other two and the latter two are not exactly linearly separable from each other [10].

Chen and Liu [10] applied visual rendering to the Iris dataset. Fig. 4.4 shows their clustering results for the Iris dataset. The VISTA system that they used implements a linear and reliable mapping model to visualize the *k*-dimensional dataset in a 2D star-coordinate space. It allows users to validate and interactively refine the cluster structure based on their visual experience as well as on their domain knowledge. They found that one cluster had been separated from the other two. The gap between clusters A and B can be visually perceived but is not very clear. Fig. 4.4 explains why two is the number of clusters in our cluster validity analysis results. Cole also conducted tests on the Iris dataset using general genetic algorithms [12]. The values of the main parameters he used in the genetic algorithm are: number of iterations = 1000, range of exponential mutation rate = from 10.0 to 0.000001, population size = 50, crossover probability = 1.00. For the cluster validity, the optimal number of clusters obtained are 3 for the Davies Bouldin method and 2 for the Calinski and Harabase method.

The clustering approach described in this paper was run 10 times with the following parameters: population size = 100, *t_dom* (number of comparison set = 10), crossover = 0.8, and mutation = 0.01. Threshold = 0.0001 was used to check if the population stops evolution after 50 generations or if the process needs to be stopped. In addition, the range of [1,10] was picked for finding the optimal number of clusters for the experiments, which is the same as for the Ruspini dataset.

Average changes in the Pareto-optimal front by running the proposed algorithm for the Iris dataset are displayed in Fig. 4.5 for different generations. It demonstrates how the system converges to an optimal Pareto-optimal front. As the actual change

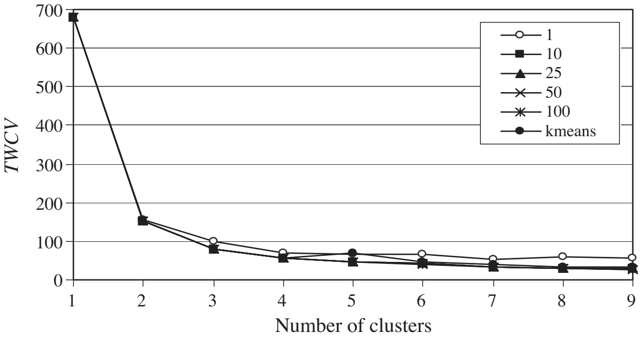

Fig. 4.5. Pareto-fronts for IRIS dataset.

Table 4.2
Iris dataset *TWCV* for $k = 6$ and $k = 9$.

Iteration	*TWCV*(6)	TWCV(9)
1	65.9482	57.2637
10	41.708	29.2061
25	41.708	28.3555
50	41.708	28.1758
100	39.043	28.1758
k-means	45.5185	34.1203

in the value of *TWVC* is not reflected in the curves plotted in Fig. 4.5 (the curves almost overlap), some key *TWVC* values are reported in Table 4.2.

The obtained results were tested and analyzed for the Iris dataset using the six indexes mentioned before. The average results of 10 runs are reported in Figs. 4.6 and 4.7. Finally, the results obtained are compared with the corresponding results reported by the other researchers [10,12]. According to [10], the optimal number of clusters found for the Iris data is 3, which ranks second for all the indexes except S-Dbw and C index (see Figs. 4.6 and 4.7). This finding is consistent with the result of the DB cluster validity index published by Cole [12]. The reason that these clusters are not the best is that the good values of the six indices indicate "good" clustering, which includes properly combined compactness and separation. Clusters are more compact but less separate from each other for the number of clusters taken as 3, while clusters with number of clusters taken as 2 are better separated. The visual clustering results given by Chen and Liu [10] show this difference clearly. The C index is likely to be data dependent and the behavior of the index may change when different data structures are used as reported in [18].

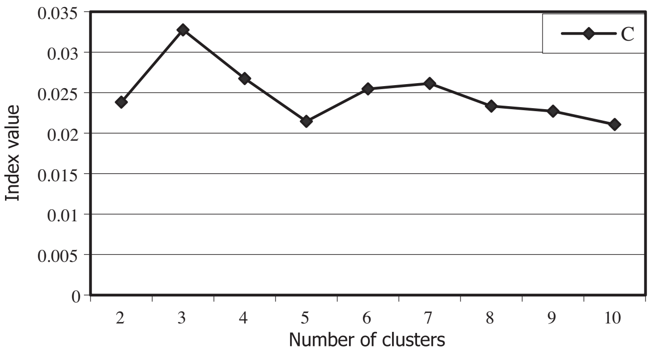

Fig. 4.7. Iris dataset cluster validity results using C index.

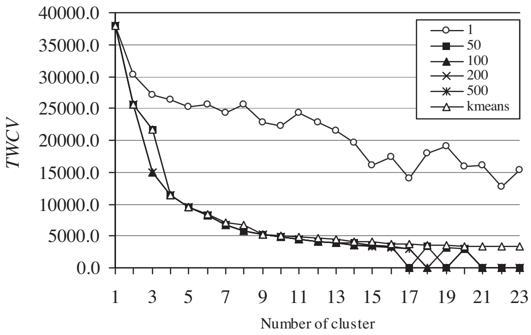

Fig. 4.8. Pareto-fronts for Fig2data dataset.

4.3. The Fig2data dataset

The Fig2data dataset is the time course of serum stimulation of primary human fibroblasts. It contains the expression data for 517 genes of which the expression changed substantially in response to serum. Each gene has 19 expressions ranging from 15 min to 24 h [10,27].

Lu et al. [38] applied the Fast Genetic K-means Algorithm to Fig2data. They selected mutation probability = 0.01, population size = 50, and generation = 100 as their parameter setting and obtained a fast clustering process.

The multi-objective genetic algorithm-based approach MOKGA described in this paper has been applied to the Fig2data dataset. Experiments were conducted with the following parameters: population size = 150, *t_dom* (number of comparison set = 10) and crossover = 0.8, mutation = 0.005, gene mutation rate = 0.005, and threshold = 0.0001 forces the stopping condition in case reached before the evolution reaches the ultimate stopping condition of

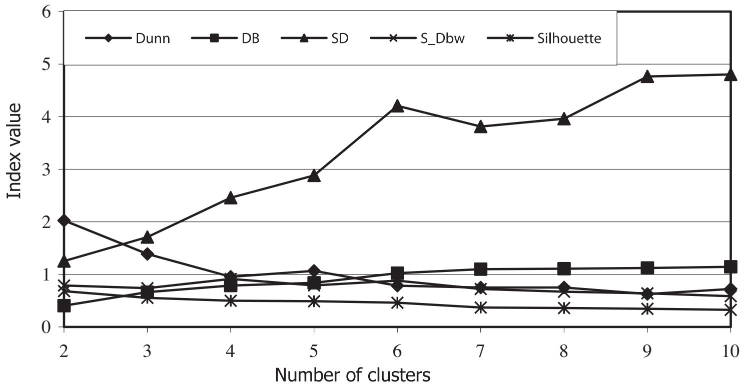

Fig. 4.6. Iris dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices.

Table 4.3
Fig2data dataset *TWCV* for *k* = 16.

Iteration	TWCV
1	17406.3
50	3371.91
100	3303.5
200	3303.21
300	3214.34
400	3211.25
500	3202.04
k-means	3803.62

50 generations. The range of [1,25] was picked to find the optimal number of clusters.

The corresponding experimental results are demonstrated in Fig. 4.8 and Table 4.3. They also show how the system quickly converges to an optimal Pareto front; the generations almost overlap after the 50th generation. As shown in Table 4.3, the variation in the TWCV is very small. Figs. 4.9 and 4.10 report validity results and reflect comparisons with the studies described in the literature [27,38]. The study of Iyer et al. [27] shows the optimal number of clusters for Fig2data as 10. Consistently, the results in this paper indicate that 10 ranks among the best results for the C index, and 10 clusters is among the best for other indices. According to Halkidi et al. [18], SD, S_Dbw, DB, Silhouette, and Dunn indices cannot properly handle arbitrarily shaped clusters, so they do not always give satisfactory results.

4.4. The NCI60 cancer dataset

The NCI60 dataset is a gene expression database for the molecular pharmacology of cancer. It contains 728 genes and 60 cell lines derived from cancers of colorectal, renal, ovarian, breast, prostate, lung, and central nervous system origin, leukemia and melanoma. Growth inhibition is assessed from changes in total cellular protein after 48 h of drug treatment using a sulphorhodamine B assay. The patterns of drug activity across the cell lines provide information on mechanisms of drug action, resistance, and modulation [58]. In the clustering test reported in this paper, there is a need to test cell–cell correlations on the basis of drug activity profiles, which are the gene expression data available.

The study by Scherf et al. [58] uses an average-linkage algorithm and a metric based on the growth inhibitory activities of the 1400 compounds for the cancer dataset. The authors observed 15 distinct branches at an average inter-cluster correlation coefficient of at least 0.3. In this method, the correlation parameter was used to control the clustering results. It might be hard to decide if it is an unsupervised clustering task.

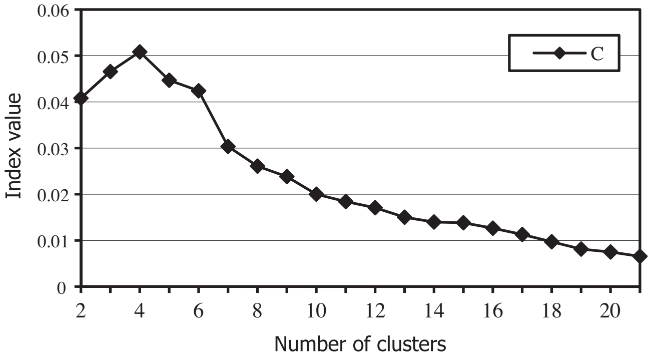

Fig. 4.10. Fig2data dataset cluster validity results using C index.

The multi-objective genetic algorithm-based approach MOKGA described in this paper has been run for the NCI60 cancer dataset with the following parameters: population size = 100, *t_dom* (number of comparison set = 10) and crossover = 0.8, mutation = 0.005, gene mutation rate = 0.005, and threshold = 0.0001 which is used to check if the population stops evolution for 50 generations and if the process needs to be stopped. The range of [1,20] was picked to find the optimal number of clusters.

Changes in the Pareto-optimal front after running the algorithm are displayed in Fig. 4.11 and Table 4.4 for different generations. The reported changes demonstrate how the system converges to an optimal Pareto-optimal front.

Figs. 4.12 and 4.13 show the average results obtained. For the cancer (NCI60) dataset, we have 15 in the Pareto optimal front; this value also ranks the sixth for DB index, fifth for SD index and fifth for the C index. These are consistent with the results reported in [58]. Some indices values are not good because index values are highly dependent on the shape of the clusters. This justifies the need to apply multiple indices and majority voting in order to eliminate the bias of distorted indices.

4.5. The Leukemia dataset

The third microarray dataset used in this paper is the Leukemia dataset, which has 38 acute leukemia samples and 50 genes. The

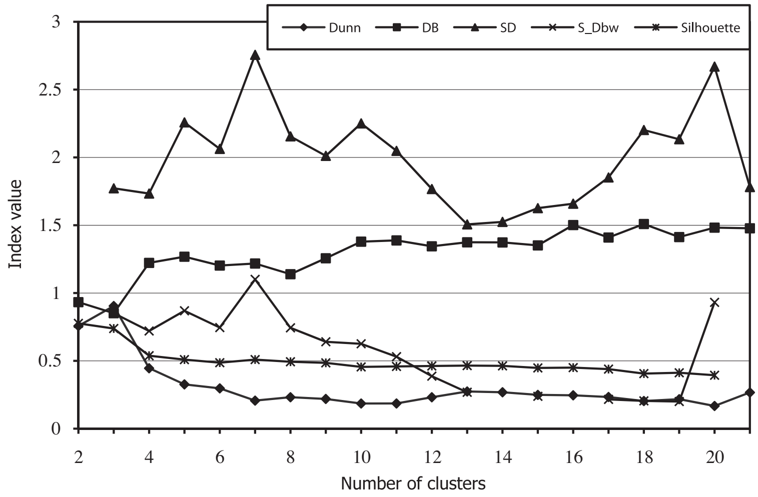

Fig. 4.9. Fig2data dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices.

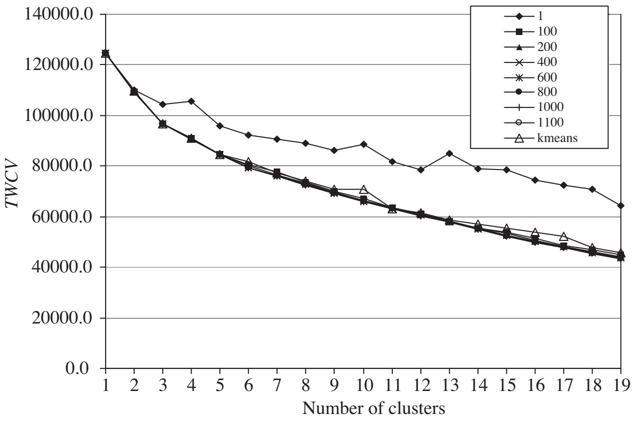

Fig. 4.11. Pareto-fronts for Cancer dataset.

Table 4.4
Cancer dataset *TWCV* for *k* = 16.

Iteration	TWCV
1	78435.2
100	53785
200	53210.5
400	52571.8
600	52571.8
800	52398.1
1000	52398.1
1100	52385.3
k-means	53673.2

purposes of the testing include clustering cell samples into groups and finding subclasses in the dataset.

The study by Golub et al. [17] uses Self-Organizing Maps (SOMs) to group the Leukemia dataset. In this approach, the user specifies the number of clusters to be identified. SOM finds an optimal set of "centroids" around which the data points appear to aggregate. It then partitions the data set with each centroid defining a cluster consisting of the data points nearest to it. Golub [17] got two clusters acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL), as well as the distinction between B-cell and T-cell ALL, i.e., that the optimal number of clusters is 2 or 3 (with subclasses).

The multi-objective genetic algorithm-based approach described in this paper was run for the Leukemia dataset with the following parameters: population size = 100, *t_dom* (number of comparison set = 10) and crossover = 0.8, mutation = 0.005, gene mutation rate = 0.005, and threshold = 0.01 for the possibility of stopping the evolution before reaching 50 generations. The range of [1,10] was picked for finding the optimal number of clusters.

Changes in the Pareto-optimal front are displayed in Fig. 4.14 and Table 4.5 for different generations. The results demonstrate how the system converges to an optimal Pareto-optimal front.

The Leukemia dataset clustering results shown in Figs. 4.15 and 4.16 indicate the same conclusions reported by Golub et al. [17]. They also indicate that 2 (AML and ALL) is the best number of clusters after the validity analysis with Dunn index, DB index, SD index, and Silhouette and 3 (AML, B-cell ALL and T-cell ALL) is the second best. C index shows that 2 is the best number of clusters and 3 is the second best.

It can be seen from Fig. 4.15 that S_Dbw is an exception. The SD index gives good values but S_Dbw does not. This indicates that the inter-cluster density for number of clusters taken as 2 or 3 is not

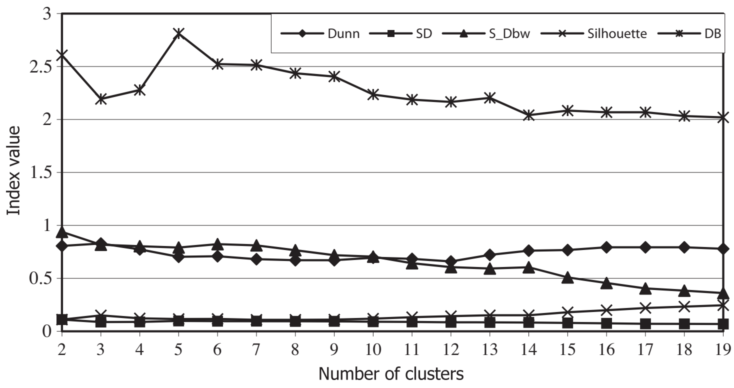

Fig. 4.12. Cancer dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices.

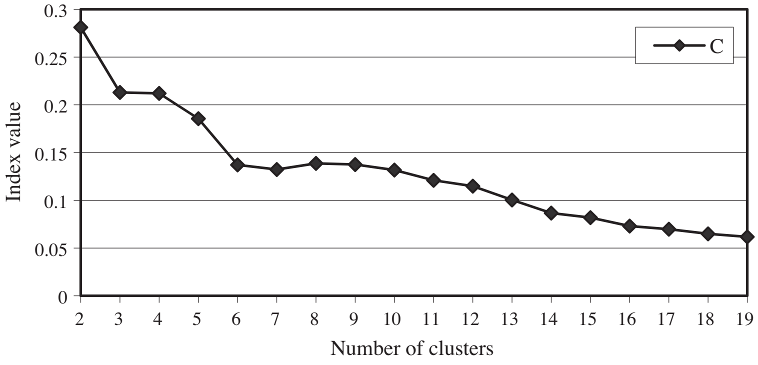

Fig. 4.13. Cancer dataset cluster validity results using C index.

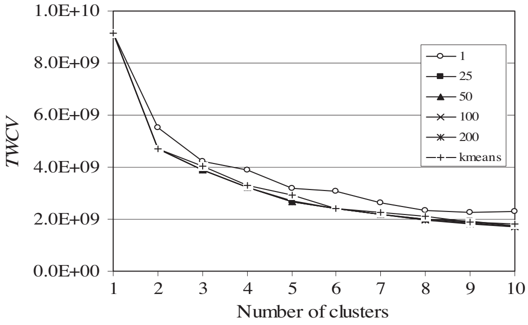

Fig. 4.14. Pareto-fronts for Leukemia dataset.

Table 4.5
Leukemia dataset *TWCV* for $k = 9$.

Iteration	TWCV
1	2.25E+09
25	1.94E+09
50	1.88E+09
100	1.84E+09
200	1.81E+09
k-means	1.88E+09

high for the 38 samples. Experimental results in this paper also indicate that the S_Dbw index is not suitable to test small datasets with fewer than 40 instances.

4.6. Breast cancer datasets

In this section, we apply the MOKGA algorithm to cluster breast cancer microarray data, since breast cancer is known to be a heterogeneous class of cancer, i.e., classification of genes/tumors is generally unstable. We have chosen two microarray datasets for this purpose: GSE12093 [71] and GSE9195 [37], available at http://www.ncbi.nlm.nih.gov/geo/.

4.6.1. The GSE12093 dataset

The GSE12093 dataset has 76-gene signatures defining high-risk patients that benefit from adjuvant tamoxifen therapy, from 136 breast cancer samples that were treated with tamoxifen. It contains 22,284 genes with 136 attributes/features. We use filtering standard of more than 200% coefficient of variation to reduce the data size and the distribution of this dataset is not sensitive to standard deviation or other filtering criteria.

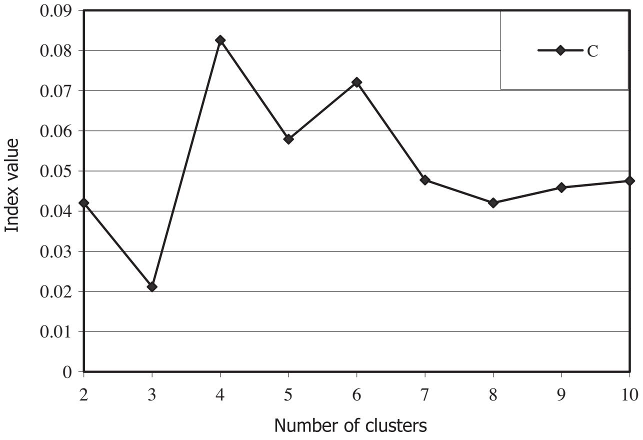

Fig. 4.16. Leukemia dataset cluster validity results using C index.

The multi-objective genetic algorithm-based approach proposed in this paper was run ten times with the following parameters: population size = 150, *t_dom* (the number of comparison set = 10) and crossover = 0.8 and mutation = 0.01. Threshold = 0.1 has been used to check if the population stops evolution for 50 generations and if the process needs to be stopped. The range of [1,6] was picked for finding the optimal number of clusters.

Changes in the Pareto-optimal front by running the algorithm for the GSE12093 dataset are displayed in Fig. 4.17 for different generations to demonstrate the rate of convergence of the algorithm to an optimal Pareto-optimal front. The actual change in the value of *TWVC* is not reflected in Fig. 4.17 where the values are very close and all the five curves almost overlap due to the scale used.

We performed cluster validity analyses on the filtered GS12093 datasets to compare the results of our experiments. We used three indices from internal measures (connectivity, Dunn and Silhouette index) and four from stability measures (Average proportion of non-overlap (APN), Average distance (AD), Average distance between means (ADM) and Figure of merit (FOM)). The test results are reported in Figs. 4.18 and 4.19 for internal measures indices and stability measures indices, respectively. All the three internal measures indices and the two stability measures indices show the same results, with similar trend.

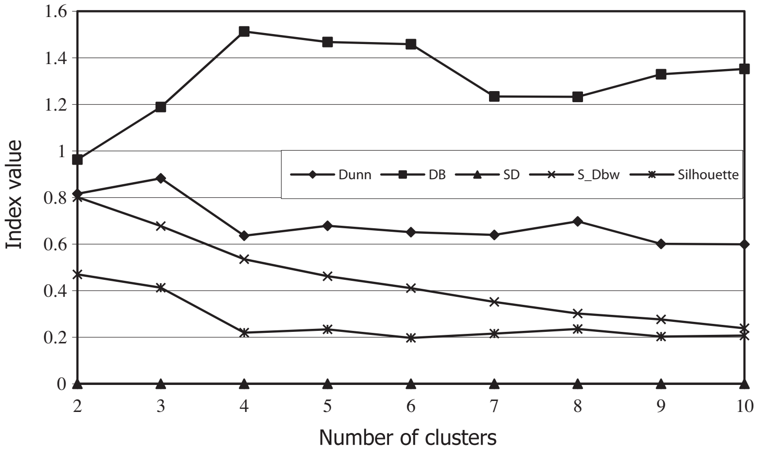

Fig. 4.15. Leukemia dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices.

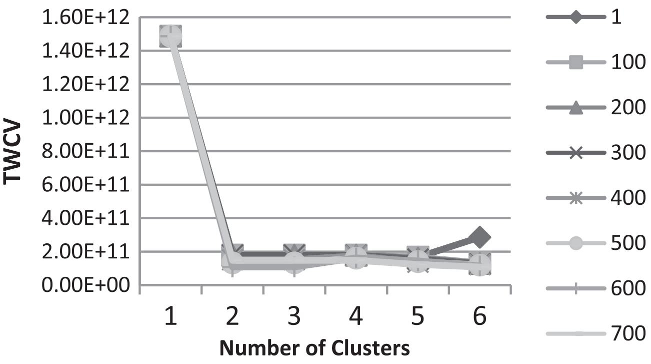

Fig. 4.17. Pareto-fronts for GSE12093 dataset.

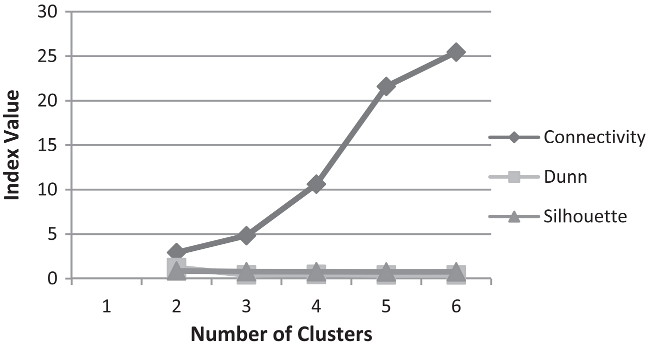

Fig. 4.18. GSE12093 dataset cluster validity results using Connectivity, Dunn and Silhouette indices.

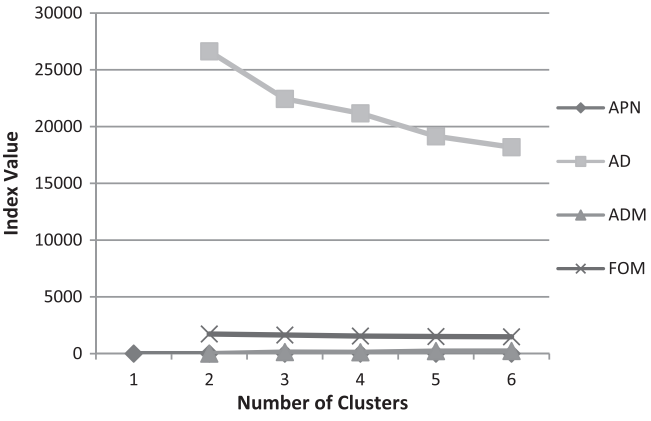

Fig. 4.19. GSE12093 dataset cluster validity results using stability measures.

4.6.2. The GSE9195 dataset

The GSE9195 dataset contains molecular profiling in estrogen receptor-positive (ER+) breast cancer treated with tamoxifen. Gene expression profiling is used to develop an outcome-based predictor using a training set of 255 ER + BC samples. The data set contains 54,675 samples with 77 attributes/features. We filtered out data having standard deviation value over 1.6 in order to reduce the data size.

The multi-objective genetic algorithm-based approach proposed in this paper was run ten times with the following parameters: population size = 150, *t_dom* (the number of comparison set = 10) and crossover = 0.8 and mutation = 0.01. Threshold = 0.1 has been used to check if the population stops evolution for 50 generations and if the process needs to be stopped. The same range for the number of clusters is used.

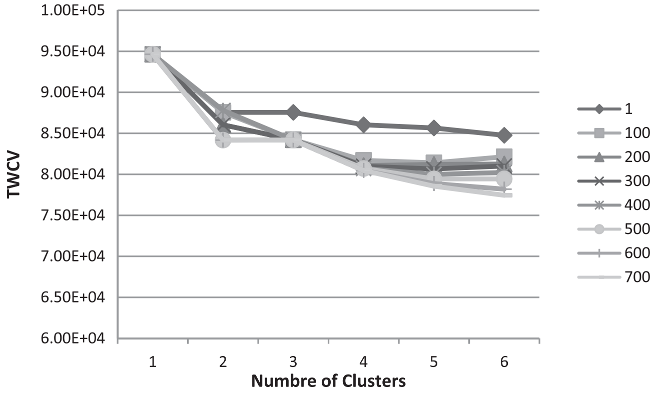

Fig. 4.20. Pareto-fronts for GSE9195 dataset.

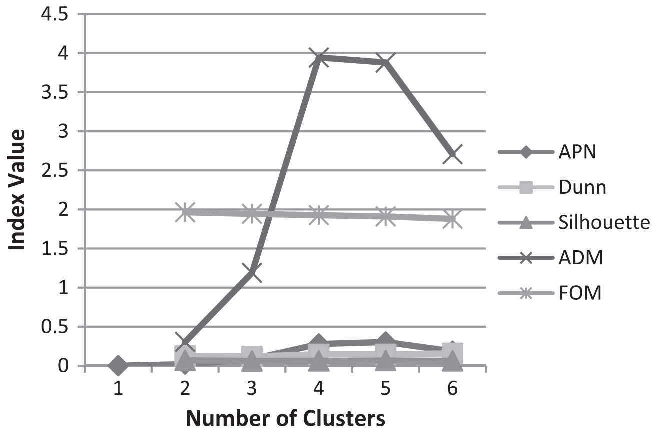

Fig. 4.21. GSE9195 dataset cluster validity results using APN, Dunn, ADM, FOM and Silhouette indices.

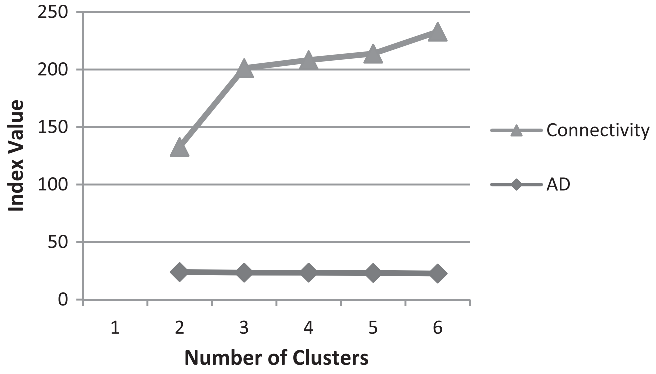

Fig. 4.22. GSE9195 dataset cluster validity results using connectivity and AD.

Changes in the Pareto-optimal front by running the algorithm for the GSE9195 datasets are displayed in Fig. 4.20 for different generations. The actual change in the value of *TWVC* is not reflected in Fig. 4.20 where the values are very close and all the five curves almost overlap due to the scale used.

We run for the GSE9195 dataset the same validity process described in the previous section. However, due to the large variances of the index values, we re-grouped the indices and show them in two figures. Fig. 4.21 shows the indices with values between 0 and 6, while Fig. 4.22 shows the connectivity and AD indices with larger value.

4.7. General evaluation and comparisons with other methods

As discussed in the previous section, experiments were conducted to examine convergence and performance of the proposed MOKGA clustering system using seven datasets. In this section, a general evaluation is given, and the MOKGA system is compared with other methods on the basis of the results reported by the other researchers who used the same datasets.

The Ruspini dataset clustering result shows that four is the optimal number of clusters reported by all the cluster validity analysis indexes. This is consistent with earlier results, e.g., [56]. The Iris dataset gives similar result with the solutions of having the number of clusters two as the best solution and 3 clusters as the second best solution; both values are acceptable and have been reported by other researchers separately. According to the work described in [27], Fig2data has 10 clusters. The proposed approach gave the same result using the C index clustering validity method. The utilized cancer data has 15 clusters according to the result reported in [58]. MOKGA produces the same result using the DB index. The optimal number of clusters of the Leukemia dataset as agreed upon in the literature is 2 or 3 (with subclasses). Fortunately, MOKGA reported the same results using Dunn, DB, SD, and Silhouette indices.

All the results we have reported for the seven datasets are consistent with the counterparts reported in the literature. These results highly emphasize MOKGA as a powerful clustering approach that can be successfully applied to various application domains.

4.7.1. MOKGA vs. Fast Genetic K-mean Algorithm (FGKA)

Since MOKGA has been developed on the basis of Fast Genetic K-mean Algorithm (FGKA) [38] and Niched Pareto Genetic Algorithm (NPGA), MOKGA and FGKA share many features: both are evolutionary algorithms; they have the same mutation and *K*-mean operators; and they both use *TWCV* for the fitness value evaluation.

According to the results, MOKGA and FGKA got similar TWCV values, MOKGA obviously needs more generations to reach the stable state, this might be because MOKGA is using separateness of clustering as another measure for checking the quality of the results and it is optimizing chromosomes with different number of clusters altogether.

MOKGA has some advantages over FGKA and GKA: it can find Pareto optimal front, which allows us to get an overview of the entire clustering possibilities and to get the optimal clustering results in one run; it does not need the number of clusters as a parameter, which is very important because clustering is an unsupervised task, and we usually do not have any idea about the number of clusters before the clustering process is completed. These two issues are real concerns for FGKA, GKA and most of the other clustering algorithms.

4.7.2. MOKGA vs. K-means algorithm

Both MOKGA and the K-means algorithm minimize the overall within-cluster dispersion by iterative reallocation of cluster members. MOKGA has some advantages over K-means: it can find Pareto optimal front; it does not need the number of clusters as a parameter; MOKGA can find global optimal solutions by applying mutation and crossover operators on surviving intermediate solutions. MOKGA combines both advantages of the genetic algorithm and advantages of K-means: by using GA operators it can get global optimal solutions, by using k-means operators MOKGA can get solutions faster.

4.7.3. MOKGA vs. Neighborhood analysis

The study described in [17] uses SOM to group instances in the Leukemia dataset. Their method reported 2 classes, and for each of them, they got 2 subclasses. Exactly the same results are obtained in the study described in this paper except for the S_Dbw index. Experimental results reported in this paper indicate that the S_Dbw index is not suitable to test small datasets, like when the number of instances is less than 40. In the experiment conducted for the study described in [17], they used the SOM method with user defined number of clusters, whereas the method proposed in this paper does not need such value to be predefined.

4.7.4. MOKGA vs. Average-linkage algorithm

The study described in [58] uses an average-linkage algorithm and a metric based on the cancer dataset. A correlation parameter was applied to control the clustering results. For the case of an unsupervised clustering task, this parameter might be difficult to decide on even by domain experts. The number of clusters 15 was obtained in this paper. It ranks the first for overall performance in the DB index. This is consistent with the result reported in [58].

4.7.5. MOKGA vs. Visual rendering

Keke Chen applied visual rendering clustering algorithm on the Iris dataset. The system implements a linear mapping model to visualize *k*-dimensional data sets in a 2D star-coordinate space; then it provides a set of interactive rendering operations to enable users to validate and interactively refine the cluster structure based on their visual experience as well as their domain knowledge. Using this method, Chen successfully divided the data set into three clusters. But, this system needs manual parameter adjustment to get a better separate map and manual boundary set. These are inefficient and may cause some errors. Without needing such manual process, MOKGA successfully grouped the data set into three clusters. Results clearly show that separating them into two clusters is also reasonable. This can be verified from the map delivered by the visual rendering method. In comparison to the visual rendering method, MOKGA has the following advantages: it is more efficient in the sense that no user's input is required during the clustering process, and it also can give users a more clear cluster validity result so that users can get an overview about the dataset. But, the visual rendering method has the advantage that users can get a visual clustering result and it may work well in dealing with clusters of irregular shapes. We have a plan to extend MOKGA with a visual interface which will be capable of displaying the alternative clustering solutions and how they evolve during the genetic algorithm process.

4.7.6. MOKGA vs. Genetic Clustering Algorithm (GCA)

Rowena Marie Cole [56] used a genetic algorithm (GCA) for clustering the Ruspini dataset. We got the same clustering result they reported. Rowena's clustering system is similar to the system proposed in this paper, they both have evolutionary based clustering algorithm and clustering validity methods; but GCA cannot find Pareto optimal front in one run; they find one solution per run which is time and effort consuming. Further, the process is relatively complex. Even if various solutions are reported by a number of runs, there is no guarantee that the individual solutions will be as compact as the counterparts produced along the pareto-optimal front reported by MOKGA.

5. Discussions

This paper investigates the clustering approaches in general and highlights their applicability for clustering datasets from various application domains, including gene expression datasets [17,36]. The covered approaches include hierarchical clustering [21], partitional clustering [33], graph-based [4] and model-based [3,70] approaches.

A multi-objective genetic algorithm called MOKGA is described in this paper to handle the data clustering problems. It is developed on the basis of the Niched Pareto optimal and fast *K*-means genetic algorithm. By using MOKGA, the main target is finding the Pareto-optimal front sought to help the user to have accessibility to 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. The applicability and effectiveness of the developed clustering approach are demonstrated by conducting experiments using the seven datasets from various domains, namely figure 2data, cancer (NCI60) and Leukemia, two breast cancer datasets, Iris and Ruspini.

In MOKGA, both crossover and mutation operators are used for the evolutionary process in addition to the *K*-means operator which is applied 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 compared with the threshold, which can be used to decide when to terminate the genetic process.

MOKGA overcomes the difficulty of determining the weight of each objective function by taking part in the fitness when dealing with this multiple objectives problem. Otherwise, the user would have been expected to do many trials with different weighting of objectives as in traditional genetic algorithms. This method also gives the user an overview of different number of clusters, which may help them in finding subclasses and optimal number of clusters in a single run, whereas traditional methods like SOM, *K*-means, hieratical clustering algorithms and GCA cannot find optimal number of clusters or need it as a predefined parameter.

MOKGA is less susceptible to the shape or continuity of the Pareto front. It can easily deal with discontinuous or concave Pareto fronts. These two issues are real concerns for mathematical programming techniques, like model-based approaches such as Bayesain method and mixed model-based clustering algorithms.

6. Conclusions

There are some possible areas of improvement for MOKGA. In this paper, cluster validity techniques, including Silhoutte, C index, Dunn's index, DB index, SD index and S-Dbw index, were used to evaluate the solutions in the Pareto optimal front and to get the optimal number of clusters. The overall performance is good, but it can be seen that S_Dbw index is more suitable for evaluating large datasets than small ones. Hence, choosing suitable index to get the optimal number of clusters will be an issue in the clustering process, especially when there are arbitrarily shaped clusters. Other future research directions include the application of MOKGA to other microarray clustering problems, such as biclustering problems [13], or using other criteria to test cluster validity. Further, the current version of MOKGA as presented in this paper does support crisp clustering and it is not capable of identifying outliers. Realizing these as vital areas of research for clustering algorithms, we plan to turn MOKGA into a comprehensive solution that can move forward from the alternative solutions into three main directions. First, we want to benefit from the alternative solutions to produce a fuzzy clustering solution. Second, we want to be able to identify outliers by employing information from the various alternative solutions along the Pareto-optimal front. Finally, we will also investigate the possibility of producing a unique more compact solution by considering the clusters reported from various solutions along the Pareto front.

References

[1] S. Bandyopadhyay, A. Mukhopadhyay, U. Maulik, An improved algorithm for clustering gene expression data, Bioinformatics 23 (21) (2007) 2859–2865.

[2] Y. Barash, Context-specific Bayesian clustering for gene expression data, J. Computat. Biol. 9 (2002) 169–191.

[3] Y. Barash, N. Friedman, Context-specific Bayesian clustering for gene expression data, in: Fifth Annual Inter. Conf. on Computational Molecular Biology (RECOMB 2001), 2001, pp. 12–21.

[4] A. Ben-Hur, A. Elisseeff, I. Guyon, A stability based method for discovering structure in clustered data, in: Proc. of Pacific Symposium on Biocomputing (PSB), 2002, pp. 6–17.

[5] A. Ben-Dor, R. Shamir, Z. Yakhini, Clustering gene expression patterns, J. Computat. Biol. 6 (3–4) (1999) 281.

[6] A. Ben-Hur, I. Guyon, Detecting stable clusters using principal component analysis, in: M.J. Brownstein, A. Kohodursky (Eds.), Methods in Molecular Biology, Humana Press, 2003, pp. 159–182.

[7] N. Bolshakova, F. Azuaje, Improving expression data mining through cluster validation, in: Proc. of IEEE Conference on Information Technology Applications in Biomedicine, 2003, pp. 19–22.

[8] A. Brazma, A. Robinson, G. Cameron, M. Ashburner, One-stop shop for microarray data, Nature 403 (6771) (2000) 699–700.

[9] A. Charalambides, Distributions of random partitions and their applications, Methodol. Comput. Appl. Probab. 9 (2) (2007) 163–193.

[10] K. Chen, L. Liu, Validating and refining clusters via visual rendering. Gene expression data of the genomic resources, in: International Conference on Data Mining (ICDM), 2003, pp. 501–504.

[11] Y. Chi, X. Song, D. Zhou, K. Hino, B.L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. in: Proc. International Conference on Knowledge Discovery and Data Mining (KDD'07), 2007, pp. 153–162.

[12] R.M. Cole, Clustering with Genetic Algorithms, 1998. http://www.cs.uwa.edu.au/pub/robvis/theses/RowenaCole.

[13] K. Curtis, M. Brand, Control analysis of DNA microarray expression data, Mol. Biol. Rep. 29 (1–2) (2002) 67–71.

[14] D. Datta, J.R. Figuera, C.M. Fonseca, F. Tavares-Pereira, Graph partitioning through a multi-objective evolutionary algorithm: a preliminary study, in: Proc. of the Genetic and Evolutionary Computation Conference (GECCO'08), 2008, pp. 625–632.

[15] E. Domany, Cluster analysis of gene expression data, Physics 110 (2002) 11–17.

[16] F. Folino, C. Pizzuti, A multiobjective and evolutionary clustering method for dynamic networks, in: Proc. of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2010), Odense, Denmark, August 2010, pp. 256–263.

[17] T.R. Golub et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[18] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Clustering Validity Checking Methods: Part II. Special Interest Group on Management of Data (SIGMOD) Record, vol. 31, no. 3, 2002, pp. 19–27.

[19] J. Handl, J.D. Knowles, An evolutionary approach to multiobjective clustering, in: IEEE Trans. Evolutionary C, IEEE Conference on Evolutionary Computation, Piscataway, NJ, vol. 1, 1994, pp. 82–87.

[20] S. J. Harendra, A Review of DNA Microarray Data Analysis, Biochemistry 218/Medical Information Sciences, 231, 2002.

[21] J.A. Hartigan, Clustering Algorithms, John Wiley and Sons, New York, 1975. pp. 353.

[22] J. Horn, N. Nafpliotis, D.E. Goldberg, A niched pareto genetic algorithm for multiobjective optimization, Proc. Comput. 11 (1) (2007) 56–76.

[23] N. Hoshino, Random clustering based on the conditional inverse Gaussian–Poisson distribution, J. Jpn. Statist. Soc. 33 (1) (2003) 105–117.

[24] E.R. Hruschka, Ricardo J.G.B. Campello, A.A. Freitas, A.C.P.L.F. de Carvalho, A survey of evolutionary algorithms for clustering, IEEE Trans. Syst., Man, Cybern. C: Appl. Rev. 39 (2) (2009).

[25] T.R. Hughes et al., Functional discovery via a compendium of expression profiles, Cell 102 (2000) 109–126.

[26] M. Inaba, H. Imai, N. Katoh, Experimental results of randomized clustering algorithm, in: SCG '96: Proceedings of the Twelfth Annual Symposium on Computational Geometry, ACM, New York, NY, USA, 1996, pp. 401–402.

[27] V.R. Iyer et al., The transcriptional program in the response of human fibroblasts to serum, Science 283 (5398) (1999). pp. 83–87.

[28] D. Jiang, C. Tang, A. Zhang, Cluster analysis for gene expression data: a survey, IEEE Trans. Knowl. Data Eng. (2003) 1–5.

[29] S. Kaski, Data exploration using Self-Organizing Maps, Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series, No. 82, March 1997, pp 57.

[30] M. Kathleen Kerr, G. Churchill, Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments, in: Proceedings of the National Academy of Sciences PNAS, 98, 2001, pp. 8961–8965.

[31] M. Kaya, R. Alhajj, Multi-objective genetic algorithms based automated clustering for fuzzy association rules mining, J. Intell. Inform. Syst. 31 (3) (2008) 243–264.

[32] K. Kianmehr, M. Kaya, A.M. ElSheikh, J. Jida, R. Alhajj, Fuzzy association rule mining framework and its application to effective classification, WIREs Data Min. Knowl. Disc. (2011), http://dx.doi.org/10.1002/widm.40.

[33] T. Kohonen, Self-Organizing Maps, Springer-Verlag, Berlin/Heidelberg, 1997.

[34] E. Levine, E. Domany, Resampling method for unsupervised estimation of cluster validity, Neural Comput. 13 (2001) 2573–2593.

[35] H. Li, Q. Zhang, Multi-objective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 12 (2) (2008).

[36] Y. Liu, T. Özyer, R. Alhajj, K. Barker, Multi-objective Genetic algorithm based clustering approach and its application to gene expression data, in: Proc. of the International Conference on Advances in Information Systems, Springer-Verlag, 2004.

[37] S. Loi, B. Haibe-Kains, C. Desmedt, P. Wirapati, et al., Predicting prognosis using molecular profiling in estrogen receptor-positive breast cancer treated with tamoxifen, BMC Genom. 9 (239) (2008).

[38] Y. Lu et al., FGKA: a fast genetic K-means clustering algorithm, in: Proc. of ACM Symposium on Applied Computing, Nicosia, Cyprus, 2004, pp. 162–163.

[39] J.C. Mar, G.J. McLachlan, Model-based clustering in gene expression microarrays: an application to breast cancer data, in: Asia-Pacific Bioinformatics Conference (APBC), 2003, pp. 139–144.

[40] N. Matake, T. Hiroyasu, M. Miki, T. Senda, Multiobjective clustering with automatic k-determination for large-scale data, in: Inter. Conf on Genetic and Evolutionary Computation Conference Companion (GECCO), 2007, pp. 861–868.

[41] U. Maulik, S. Bandyopadhyay, A. Mukhopadhyay, Multiobjective Genetic Algorithms for Clustering – Applications in Data Mining and Bioinformatics, Springer, 2011.

[42] U. Maulik, A. Mukhopadhyay, S. Bandyopadhyay, Combining pareto-optimal clusters using supervised learning for identifying co-expressed genes, BMC Bioinform. 10 (27) (2009).

[43] P. McConnell, K. Johnson, D.J. Lockhart, An introduction to DNA microarrays, in: Proc. of Critical Assessment of Massive Data Analysis (CAMDA), 2001.

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

[45] B.J.T. Morgan, A.P.G. Ray, Non-uniqueness and inversions in cluster analysis, Appl. Stat. 44 (1) (1995) 117–134.

[46] U. Möller, D. Radke, F. Thies, Testing the significance of clusters found in gene expression data, in: Proc. of European Conference on Computational Biology, Paris, 2003, pp. 26–30.

[47] A. Mukhopadhyay, U. Maulik, Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with SVM classifier, IEEE Trans. Geosci. Remote Sens. 47 (4) (2009) 1132–1138.

[48] M. Neef, D. Thierens, H. Arciszewski, A case study of a multi-objective elitist recombinative genetic algorithm with coevolutionary sharing, in: P. Angeline (Ed.), Proc. of the International Congress on Evolutionary Computation, Priscataway, 1999, pp. 796–803.

[49] T. Özyer, R. Alhajj, Deciding on number of clusters by multi-objective optimization and validity analysis, J. Multiple-Valued Logic Soft Comput. 14 (3–5) (2008) 457–474.

[50] T. Özyer, R. Alhajj, Parallel clustering of high dimensional data by integrating multi-objective genetic algorithm with divide and conquer, Appl. Intell. 31 (3) (2009) 318–331.

[51] T. Özyer, M. Zhang, R. Alhajj, Integrating multi-objective genetic algorithm based clustering and data partitioning for skyline computation, Appl. Intell. 35 (1) (2011) 110–122.

[52] V. Pareto, Cours d'economic politique, Dronz, Geneva, Switzerland, 1896.

[53] M. Ramoni, P. Sebastiani, I.S. Kohane, Cluster analysis of gene expression dynamics, Proc. Natl. Acad. Sci. 14 (2002) 9121–9126.

[54] V. Roth, T. Lange, M. Braun, M. Buhmann. A Resampling Approach to Cluster Validation. Computational Statistics (COMPSTAT), Physica Verlag, 2002, pp. 123–128.

[55] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.

[56] E.H. Ruspini, Numerical methods for fuzzy clustering, Inform. Sci. 2 (1970) 319–350.

[57] S. Saha, S. Bandyopadhyay, A new symmetry based multiobjective clustering technique for automatic evolution of clusters, Pattern Recogn. 43 (3) (2010) 738–751.

[58] U. Scherf et al., A gene expression database for the molecular pharmacology of cancer, Nat. Genetic 24 (2000) 236–244.

[59] E. Segal, D. Koller, Probabilistic hierarchical clustering for biological data, in: Proc. Inter. Conf. on Research in Computational Molecular Biology, Washington, DC, April 2002, pp. 273–280.

[60] S. Selim, M. Ismail, K-means-type algorithms: a generalized convergence theorem and characterization of local optimality, IEEE Trans. Pattern Anal. Machine Intell. 6 (1984) 81–87.

[61] W. Shannon, R. Culverhouse, J. Duncan, Analyzing microarray data using cluster analysis, Pharmacogenomics 4 (1) (2003) 41–52.

[62] M. Sibuya, A random clustering process, Ann. Inst. Stat. Math. 45 (3) (1993) 459–465.

[63] B. Stein, S. Meyer, F. Wissbrock, On cluster validity and the information need of users, in: Proc. of the International Conference on Artificial Intelligence and Applications, Benalmadena, Spain, September 2003.

[64] P. Tamayo et al., Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation, Proc. Natl. Acad. Sci. (1999) 2907–2912.

[65] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. Roy. Stat. Soc. JRSS-B 63 (2001) 411–423.

[66] R. Ulrich, S. Friend, Toxicogenomics and drug discovery: will new technologies help us produce better drugs?, Nat Rev. Drug Discov. 1 (2002) 84–88.

[67] P.J. Waddell, H. Kishino, Cluster inference methods and graphical models evaluated on NCI60 microarray gene expression data, Genome Inform. 11 (2000) 129–140.

[68] D. Wang, H. Ressom, M. Musavi, C. Domnisoru, Double self-organizing maps to cluster gene expression data, in: Proc. of The European Symposium on Artificial Neural Networks (ESANN), 2000, pp. 45–50.

[69] K.Y. Yeung, D.R. Haynor, W.L. Ruzzo, Validating clustering for gene expression data, Bioinformatics 17 (4) (2001) 309–318.

[70] K.Y. Yeung et al., Model-based clustering and data transformations for gene expression data, Bioinformatics 17 (2001) 977–987.

[71] Y. Zhang, A.M. Sieuwerts, M. McGreevy, G. Casey, et al., The 76-gene signature defines high-risk patients that benefit from adjuvant tamoxifen therapy, Breast Cancer Res. Treat. 116 (2) (2009) 303–309.

[72] 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, pp. 19–39.