

UNIVERSITY OF CALGARY

**Multi-objective Genetic Algorithms Based Approach to Clustering  
and Its Application to Microarray Data Analysis**

by

Yimin Liu

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES  
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE  
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

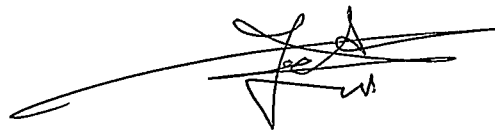
CALGARY, ALBERTA

JUNE, 2004

© YIMIN LIU 2004

UNIVERSITY OF CALGARY  
FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled " Multi-objective Genetic Algorithms Based Approach to Clustering and Its Application to Microarray Data Analysis " submitted by Yimin Liu in partial fulfilment of the requirements of the degree of Master of Science.



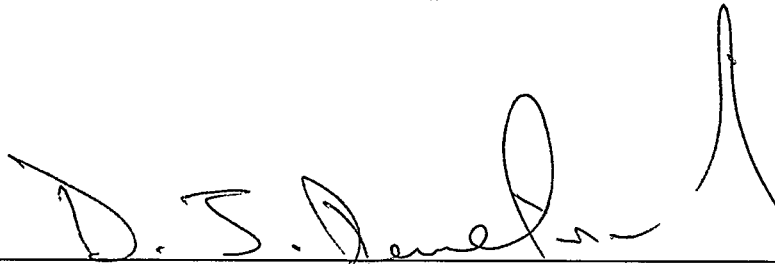
---

Supervisor, Dr. Reda Alhajj, Department of Computer Science



---

Dr. Ken Barker, Department of Computer Science



---

Dr. Douglas J. Demetrick, Department of Pathology

June 16<sup>th</sup>, 2004

Date

# Abstract

Microarray data clustering is the common methodology to analyze similar data based on expression trajectories. Clustering algorithms in general need a prior number of clusters and this is hard even for domain experts to estimate. In this thesis, a new clustering algorithm, namely the Multi-objective K-Means Genetic Algorithm (MOKGA), is proposed for clustering microarray gene expression data. After running MOKGA, a pareto-optimal front is obtained and gives the optimal number of clusters as a solution set. The obtained clustering results are then analyzed under several cluster validity techniques proposed in the literature. As a result, the optimal clusters are ranked for each validity index. In this thesis, the proposed clustering approach is tested by conducting experiments using five data sets. The obtained results are compared with those reported in the literature to demonstrate the applicability and effectiveness of the proposed approach.

## Acknowledgements

I would like to express my sincere thanks to all of those contributed in some way to the development of this research and my education.

I am particularly indebted to the supervisor Dr. Reda Alhajj for his caring guidance and continuing encouragement during the past two years and throughout the period of this research. His resourcefulness knowledge and constructive criticisms expand the limits of my own capacity.

I am also very grateful to Tansel Özyer for fruitful discussions on this thesis.

My appreciation is also extended to the remaining members of the thesis committee for their suggestions on this research.

Funding from the University of Calgary Graduate Teaching Assistantship, and the support from the secretaries in the Department of Computer Science are gratefully acknowledged.

I am also extremely grateful to my parents, husband, and sister for their encouragements, love and inspiration.

# Table of Contents

Approval Page.....	ii
Abstract.....	iii
Acknowledgements.....	iv
Table of Contents.....	v
List of Tables.....	vii
List of Figures.....	viii
Nomenclature.....	x
<b>Chapter One Introduction.....</b>	<b>1</b>
1.1 Problem definition.....	1
1.2 Motivation.....	2
1.3 Contributions.....	3
1.4 Outline of the thesis.....	4
<b>Chapter Two The Necessary Background and Related Work.....</b>	<b>6</b>
2.1 Clustering.....	6
2.1.1 What is clustering?.....	6
2.1.2 Clustering methods.....	7
2.2 Cluster validity analysis.....	11
2.2.1 Resampling techniques.....	11
2.2.2 Bootstrap method.....	12
2.2.3 Validity indexes.....	12
2.3 Microarray and its application.....	14
2.3.1 What is a microarray?.....	14
2.3.2 Applications of Microarray Technology.....	17
2.4 The usage of clustering for microarray analysis.....	19
2.5 Genetic algorithms.....	21
2.6 Multi-objective GA's.....	23
2.6.1 Multi-objective optimization.....	24
2.6.2 Multi-objective Genetic Algorithms.....	27
2.7 Related work.....	30
<b>Chapter Three The Approach.....</b>	<b>33</b>
3.1 The objectives.....	34
3.2 Chromosome encoding.....	34
3.3 Fitness evaluation and selection.....	35
3.4 Crossover and mutation.....	36
3.5 Implementation details.....	37
3.5.1 Multi-Objective Genetic <i>K</i> -means Algorithm (MOKGA).....	38
3.5.2 Cluster Validity.....	46
<b>Chapter Four Experimental Results.....</b>	<b>51</b>
4.1 The environment used for the experiments.....	51
4.2 Data sets.....	52
4.2.1 Ruspini dataset.....	52

4.2.2	Iris dataset .....	55
4.2.3	Fig2data Dataset.....	59
4.2.4	Cancer (NCI60) dataset.....	62
4.2.5	Leukaemia dataset.....	65
4.3	General Evaluation and Comparisons with Other Methods .....	68
4.3.1	General evaluation .....	69
4.3.2	Comparisons with other methods.....	69
<b>Chapter Five Discussions and Conclusions .....</b>		<b>73</b>
References.....		76
Appendix A Cluster validity results.....		87

## List of Tables

Table 4.1 Ruspini Dataset <i>TWCV</i> for $k = 8$ .....	53
Table 4.2 Iris Dataset <i>TWCV</i> for $k = 6$ and $k = 9$ .....	57
Table 4.3 Fig2data Dataset <i>TWCV</i> for $k = 16$ .....	60
Table 4.4 Cancer Dataset <i>TWCV</i> for $k = 16$ .....	64
Table 4.5 Leukaemia Dataset <i>TWCV</i> for $k = 9$ .....	67

## List of Figures

Figure 2.1 cDNA microarray schema (Taken from [DUG99]) .....	16
Figure 2.2 An example of a problem with two objective functions.....	25
Figure 2.3 Illustration of Pareto front convergence process .....	26
Figure 3.1. An example of Pareto tournament selection.....	36
Figure 3.2. Flow chart: the process of the Multi-Objective Genetic K-means Algorithm	39
Figure 4.1 Pareto-fronts for Ruspini dataset.....	53
Figure 4.2 Ruspini dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices .....	54
Figure 4.3 Ruspini dataset cluster validity results using C index .....	54
Figure 4.4. The real cluster distribution visualized with the labels from the original Iris dataset: Iris dataset clustering results from [CL03] .....	55
Figure 4.5 Pareto-fronts for IRIS dataset.....	57
Figure 4.6 Iris dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices .....	58
Figure 4.7 Iris dataset cluster validity results using C index .....	59
Figure 4.8 Pareto-fronts for Fig2data dataset .....	60
Figure 4.9 Fig2data dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices .....	61
Figure 4.10 Fig2data dataset cluster validity results using C index .....	62
Figure 4.11 Pareto-fronts for Cancer dataset .....	63



Figure 4.12 Cancer dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices .....	64
Figure 4.13 Cancer dataset cluster validity results using C index .....	65
Figure 4.14. Pareto-fronts for Leukaemia dataset.....	66
Figure 4.15 Leukemia dataset cluster validity results using Dunn, DB, SD, S_Dbw and Silhouette indices .....	67
Figure 4.16 Leukemia dataset cluster validity results using C index .....	68

# Nomenclature

$\alpha$	weighting factor
$a(i)$	average dissimilarity of $i$ -object to all other objects in the same cluster
$a_n$	gene
ALL	acute myeloid leukemia
AML	acute lymphoblastic leukemia
ANOVA	analysis of variance
$b(i)$	average dissimilarity of $i$ -object to all objects in the closest cluster
$c_i$	$i$ -cluster of certain partition
$C$	C index
$C_i$	cluster
CTWC	Coupled Two-Way Clustering
$d(c_i, c_j)$	distance between clusters $c_i$ and $c_j$ (intercluster distance)
$d(x, m_i)$	Euclidean distance between $x$ and $m_i$
$d(X_n, C_k)$	Euclidean distance between pattern $X_n$ and the centroid $C_k$ of the $k$ -th cluster
$diam(c_k)$	intracuster distance of cluster $c_k$
$d_{\max}(X_n)$	$\max_k \{d(X_n, C_k)\}$
$DB$	DB index
$Dens\_bw(nc)$	inter-cluster density
$D_{\max}$	maximum distance between cluster centers
$D_{\min}$	minimum distance between cluster centers
DSOM	Double self organizing maps

$D_{nc}$	Dunn index
$f_k(\mathbf{x})$	$k$ objective function
$F(\mathbf{x})$	$(f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$
FGKA	Fast Genetic K-means Algorithm
ID	Inter-cluster
$L$	number of pairs for calculating $S_{min}$ and $S_{max}$
$m_i$	center of cluster $C_i$ ,
MOKGA	multi-objective $K$ -means genetic algorithm
MOGA	Multi-Objective Genetic Algorithm
$n$	number of clusters.
$n_{ij}$	number of tuples that belong to the cluster $c_i$ and $c_j$
NPGA	Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
PAH	probabilistic abstraction hierarchies
$p_c$	crossover
$p_i$	probability interval of mutating gene assigned to cluster $i$
$p_k$	$k$ -th probability
$R^n$	space
$stedev$	average standard deviation of clusters.
$S$	solution or parameter space; the sum of distances over all pairs of patterns from the same cluster
$S(i)$	Silhouette index:
$S_n$	average distance of all objects from the cluster to their cluster center

$S(Q_i, Q_j)$	distance between clusters centers.
$SD(n_c)$	SD index
$S_{min}$	sum of the $l$ smallest distances if all pairs of patterns are considered
$Scat(n_c)$	average compactness of clusters
$Scatt(n_c)$	average scattering for clusters
$Scatt(n_c)$	total separation between clusters
$S_{max}$	sum of the $l$ largest distance out of all pairs.
$SF_{kd}$	sum of the $d$ -th features of all the patterns in cluster $k$ ( $G_k$ ):
SOM	Self organizing maps
TWCV	Total Within-Cluster Variation
PCA	Principal Component Analysis
$u_{ij}$	middle point of the line segment defined by the cluster centers
$v_i$	cluster centers
VEGA	Vector Evaluated Genetic Algorithm
$x_i$	optimization parameter
$\mathbf{x}$	solution set, $= (x_1, x_2, \dots, x_k)$
$X_i$	objects
$X_{nd}$	feature $d$ of pattern $X_n$
$\mathbf{y}$	solution set, $= (y_1, y_2, \dots, y_k)$
$\mathbf{y}'$	solution set, $= (y'_1, y'_2, \dots, y'_k)$
$Z_k$	number of patterns in cluster $k$ ( $G_k$ )
$\sigma(v_i)$	average standard deviation (average of the Euclidian distance between all the points) of cluster centers

$\sigma(x)$       average standard deviation of all the data points

# Chapter One

## Introduction

### 1.1 Problem definition

The central role of the DNA microarray technology in biological and biomedical domains enables researchers to observe transcription levels of many thousands of genes. Information gathered by analyzing the genes at different levels and stages of the process is used for the gene function, the reconstruction of the gene network, the diagnosis of disease conditions, and the inference of medical treatment [JGR03].

Data mining methods and techniques have a great deal of interest and application areas including bioinformatics. They are designed for extracting previously unknown significant relationships and regularities out of huge heaps of details in large data collections [SS01]. The identified gene expression levels reflecting the biological processes of interest are frequently used to analyze the inference of differentially expressed genes and their clustering. The main step in the analysis of gene expression data is to identify groups of genes/samples based on the notion of similarity. Two leading data mining tasks, classification and clustering, exhibit the capability of grouping the genes.

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 in to be able to classify them correctly.

In contrast to classification, in clustering, also known as unsupervised clustering, classes are unknown *a priori* and 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.

As described in the literature, some of the existing clustering techniques have been successfully employed in analyzing the gene expression data. These include hierarchical clustering method, 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 priori* even for experts. Thus, the problem handled in this thesis 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 some validity analysis on the resulting solutions to rank them.

## **1.2 Motivation**

When clustering microarray data without any previous knowledge about the data, it is hard to decide on the number of clusters and there are always some trade-offs between 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. This is a multi-objective problem. The solution of a multi-objective problem is a solution set, which is called a Pareto-optimal set or non-dominated set [VP1896].

In general, traditional algorithms for clustering microarray data do not produce the Pareto optimal set, and most do not lead to the optimal number of clusters in the database that they work on. For example, 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 cannot tell what is the optimal number of clusters nor give the non-dominated set, the  $K$ -means method needs the number of clusters as a predefined parameter, and it may give local optimal solutions because it is a local search from a random initial partitioning. SOM has the same disadvantage in that it requires the number of clusters as *a priori*.

Clearly, a clustering algorithm is needed in 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.

### **1.3 Contributions**

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

- Multi-objective  $K$ -means Genetic Algorithm (MOKGA) based clustering approach, which presents to the user a Pareto optimal clustering solution set without taking weight values into account. Otherwise, the user will have to consider several trials weighting with different values until a satisfactory result is obtained.



- Cluster validity analysis 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 proposed clustering approach and clustering validity analysis process are demonstrated by conducting experiments using five datasets: Fig2data, cancer (NCI60), Leukaemia data sets available at Genomics Department of Stanford University, UCI machine learning repository, Iris at Genome Research MIT, and the well-known Ruspini dataset. Finally, two papers are published from this thesis [LOAB04] [OLAB04].

#### **1.4 Outline of the thesis**

The balance of the thesis is organized as follows. Chapter 2 discusses the necessary background for this research. The concepts of clustering and microarray are introduced. Existing techniques on clustering methods, clusters validity analysis, genetic algorithms multi-objective genetic algorithms are all discussed. Other related topics including the application of microarray and the usage of clustering for microarray analysis are also covered. Finally the clustering approaches used primarily in the microarray data analysis area are reviewed.

Chapter 3 is devoted to the development of a new clustering system for clustering both gene expression and general datasets. The system has two main components: Multi-objective *K*-Means Genetic Algorithm (MOKGA) and cluster validity analysis. The purpose of MOKGA is to get the Pareto-optimal front, which gives the optimal number

of clusters as a solution set. The cluster validity analysis involves six cluster validity techniques. Methods helpful to get more optimal solutions, such as the multiple Pareto front ranking method and the Pareto front distance threshold calculating method are also proposed.

In Chapter 4, the proposed clustering system and related methods are applied to five datasets to test the applicability, performance, and efficiency of the system. Experimental results for each dataset are presented and between results from the proposed approach are compared with other similar methods.

Finally, Chapter 5 discusses the advantages and disadvantages of the proposed approaches, in comparison with other existing methods. Conclusions are made and future research directions are suggested.

## Chapter Two

### The Necessary Background and Related Work

In this chapter, the major topics necessary to understand the approach proposed in this research are discussed. These include existing clustering approaches, cluster validity analysis, microarray and its application, the usage of clustering for microarray analysis, genetic algorithm in general, and multi-objective genetic algorithms in particular.

#### 2.1 Clustering

##### 2.1.1 What is clustering?

Cluster analysis can be stated as follows: given  $N$  data points embedded in a  $D$ -dimensional space, partition the  $N$  points into  $M$  clusters, such that the points in the same cluster are “more similar” to each other than those belonging to different clusters. Through this analysis, one can identify the underlying structure of the data. A good clustering method will produce high quality clusters such that the intra-class similarity (i.e., within a cluster) is high and the inter-class similarity (i.e., between clusters) is low.

There are two kinds of clustering analysis techniques: supervised and unsupervised clustering analysis. Unsupervised methods can mine through data and extract relevant information without the presence of a teacher signal [DE02]. On the other hand, supervised methods use a teacher signal to extract information. We can say that

unsupervised methods perform the clustering job while supervised methods are more suited to the classification of datasets.

The main disadvantage of supervised methods is their limitation to hypothesis testing. Supervised methods will help accept or reject the hypothesis, but they will never reveal the unexpected and cannot lead to new hypotheses or new partitions of the data that are unexpected. And they are unable to find the mislabeled data in the training set.

Unsupervised methods, on the other hand, aim at exploring the structure of the data on the basis of correlations and similarities present in the data. In the context of gene expression, such an analysis has two goals:

1. To find groups of genes that have correlated expression profiles. The members of such a group may take part in the same biological process; and
2. To divide the samples into groups with similar gene expression profiles. Samples belonging to one group are expected to be in the same biological state. In this thesis, the method presented to accomplish these aims is called clustering and it is regarded as an unsupervised learning method.

### 2.1.2 Clustering methods

Existing clustering techniques may be classified into traditional clustering algorithms, including hierarchical clustering [JAH75], partitional clustering [TKO97], and recently emerging clustering techniques such as graph-based [BSY99] and model-based [KYY01] [YB02] approaches. Some of the existing clustering techniques have been successfully employed in analyzing the gene expression data.

### 2.1.2.1 Hierarchical clustering

Hierarchical clustering is a very well known method by biologists. A tree structure called a *dendrogram* is used to illustrate the hierarchical clustering [JTZ03]. Relationships among genes are represented by the tree using a degree of similarity symbolized with the branch lengths.

Hierarchical clustering methods are categorized into agglomerative (bottom-up) clustering and divisive (top-down) clustering. An agglomerative clustering starts with one-point (singleton) clusters and recursively merges two or more most appropriate clusters. A divisive clustering starts with one cluster of all data points and recursively splits the most appropriate cluster. For both categories, the process continues until a stopping criterion is achieved [JTZ03] [SJ02].

Hierarchical clustering algorithms have been widely used in the area of gene expression data analysis. For example, Waddell [WK00] applied hierarchical clustering method based on partial correlations on NC160 gene expression data to find a tight and closed set of genes. He then used them for graphical modeling to find the interaction of important genes of the cell cycle.

### 2.1.2.2 Partitional clustering

*K*-Means is a commonly used algorithm for partition clustering [TKO97]. 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:

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

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.

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.

### **2.1.2.3 Self organizing maps (SOM):**

Self Organizing Maps method (SOM)[SK97] is a neural network approach that uses competitive unsupervised learning and eventually the winner-takes-all approach to assign each gene to a cluster. SOMs work is somewhat like  $K$ -Means clustering but a little richer. With  $K$ -Means, one chooses the number of clusters to fit the data into; but for SOM, one chooses the shape and size of a network of clusters to fit the data into. There is one input layer and a competitive layer, so each input neuron is used for the output result of each competitive layer neuron. Two dimensional grids are used to evaluate the results. Each input neuron is connected with an arc to every neuron at the competitive layer with different weight, and competitive neurons are evaluated with an activation function. It is good because input neurons feed the competitive neurons with the varying weights in parallel by the product of perceptron learning.

Double self organizing maps (DSOM)[WRM00] 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. During the self-organizing process, both the weight and position vectors are updated. Because the position vectors are two-dimensional, we can visualize the number of groups of the position vectors by

plotting them. Thereby, we are able to determine the number of classes in the gene expression data set.

#### **2.1.2.4 Model-based clustering algorithm**

The model-based approach [RSI02] 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* [BF01] applied 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 [MM03] 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.

#### **2.1.2.5 Graph-based clustering methods**

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 same threshold similar to it [BSY99]. The study by Ben *et al* [BSY99] tries to make cliques for the clustering purpose. Examples of this approach are the Two-Way Clustering Binary tree [CB02] and the Coupled Two-Way Clustering [RSI02].

## 2.2 Cluster validity analysis

Clustering is mainly an unsupervised task, so after data clustering and data partitioning into subgroups, the validity of the result must be checked [MRT03].

### 2.2.1 Resampling techniques

Levine introduces a cluster validation method based on resampling [LD01]. The clustering algorithm is applied to each subset of the data constructed randomly. A figure of merit is proposed to identify the stable clustering solutions. The proposed procedure was tested on a one-dimensional data set, for which an analytical expression for the figure of merit was derived and compared with the corresponding numerical results.

In another paper, Roth [RLBB02] 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.

Resampling techniques have some advantages. Within the same algorithm, partitions can be attributed in the presence of noise. A slight modification of the noise may then alter the cluster structure significantly. This method controls and alters the noise by resampling the original data set. It also requires no assumption about the structure of the data, the expected clusters, or the noise in the data. Only available data is used. In addition, this method can also define an optimal number of clusters. The disadvantage of this method is that it is unsuitable for very sparse data. In this case, dilution can eliminate some of the underlying modes [LD01] [BEG02].



### 2.2.2 Bootstrap method

Bootstrapping cluster analysis begins by creating a number of simulated datasets based on statistical models, such as the analysis of variance (ANOVA) models [KC01]. For each simulated data set, a bootstrap temporal pattern can be constructed based on the estimates of the difference between genes and varieties. The filtering and clustering steps can then be repeated with these bootstrap estimates to assess the stability of the results from a cluster analysis.

This method is a straightforward way to assess the reliability of clustering results. The partition generated by an algorithm with low variability is in general more credible and therefore has high cluster validity. The disadvantage is that this method only works well when the experimental design provides enough replication.

### 2.2.3 Validity indexes

Other widely accepted criteria used by the clustering algorithms are the compactness of the clusters and their 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. Some clustering validity techniques used for the validation task include Dunn index [DUG99], Davies-Bouldin (DB) index [BRA00], Silhouette index [HUG00], C index [UF02], SD index [MJL01] and S\_Dbw index [GST99], 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 scatter (use Euclidean distance to calculate the scatter ratio) of objects within a cluster and the scatter of cluster centres. It considers the average case by using the average error of each class.

C-index is another technique being used for the cluster validity. It 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 [BA2003].

Silhouette is based on the tightness and separation. It finds the overall average of the ratio of the difference of each object's minimum average dissimilarity to all objects in other clusters.

SD index is evaluated by using the average scattering for clusters and the total scattering between clusters.

S\_Dbw is similar to SD index, but it also considers the inter-cluster density instead of the total scattering in SD, and no weighting is used. Density formula uses the average standard deviation of the clusters. The detailed formulas to calculate the indexes will be discussed in chapter 4.

Examples of other cluster validity approaches used in gene expression data analysis include Principal Component Analysis (PCA) [BG03] and Gap statistic [TWH01]. PCA is a statistical-based method that can improve the extraction of cluster structure and compare clustering solutions [BG03]. Gap statistic utilizes within-cluster distances to determine the “appropriate” number of clusters in a data set. It is good at identifying

well-separated clusters, but not for not-well-separated data and data concentrated on a subspace.

### **2.3 Microarray and its application**

Microarrays are the first tool permitting a truly integrated view of life at the molecular level. Arrays are capable of profiling patterns of expression for all mouse or human genes in a single experiment. About a quarter century ago, labelled nucleic acid molecules were found reasonable to be used to interrogate nucleic acid molecules attached to a solid support [BRA00]. Today, thousands or even tens of thousands of genes can be spotted on a microscope slide and the relative expression levels of each gene can be determined.

The development of DNA microarray technology has produced large amount of gene data through which we can monitor the expression patterns of thousands of genes under particular experimental environments and conditions. Further, we can analyze the gene information rapidly and precisely by managing them at one time.

#### **2.3.1 What is a microarray?**

An array is an ordered arrangement of samples. A DNA Microarray is an array used at the molecular level and for DNA samples with diameters less than a certain value. It provides a medium for matching known and unknown DNA samples based on base-pairing rules (i.e., A-T and G-C for DNA; A-U and G-C for RNA) and automate the process of identifying the unknowns. Microarray technology promises to monitor the whole genome on a single chip so that researchers can have a better picture of the interactions among thousands of genes simultaneously.

The general procedure of microarrays technology works as follows:

1. Single-stranded cDNA molecules are attached at fixed spots on a microarray glass slide. There may be tens of thousands of spots on an array each representing a single gene.
2. RNA from the sample and from control cells is extracted and labelled with two fluorescent labels (Cye3 and Cye5): for example, a red dye for RNA from the sample population and a green dye for that from the control population. Both extracts are washed over the microarray.
3. The gene sequences from the extracts are then hybridized to their complementary sequences in the spots. The dyes enable the amount of sample bound to a spot to be measured from the level of fluorescence emitted when excited by a laser. If the RNA from the sample population is in abundance, the spot will be red; if the RNA from the control population is in abundance, it will be green; if sample and control bind equally, the spot will be yellow; and if neither binds, it will appear black. Thus, the relative expression levels of the genes in the sample and in control populations can be estimated from the fluorescence intensities and colours for each spot [BRA00]. Figure 2.1 (taken from [DUG99]) shows the schema of this experiment.

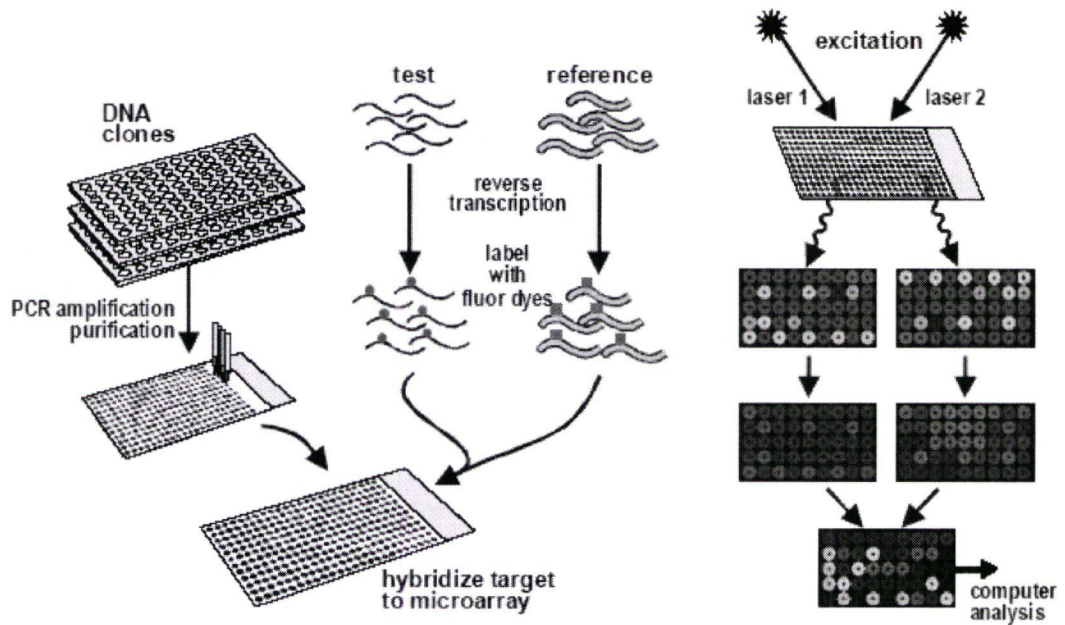


Figure 2.1 cDNA microarray schema (From [DUG99])

4. Based on the size of each DNA spot on the array, DNA arrays can be categorized as microarrays when the diameter of DNA spot is less than 250 microns and macroarrays when the diameter is larger than 300 microns. macroarrays can be easily imaged by existing gel and blot scanners while microarrays require specialized robotics and imaging equipment that are generally not commercially available as a complete system [CW03].

There are two variants of DNA microarray (also called microarray) technology that are distinguished in terms of the property the of arrayed DNA sequence with known identity:

Method 1: Probe cDNA (500~5,000 bases long) is immobilized on a solid surface. A set of targets are then added either separately or in a mixture. This method,

"traditionally" called DNA microarray, is widely considered to have been developed at Stanford University [EC99].

Method 2: An array of oligonucleotide (20~80-mer oligos) or peptide nucleic acid (PNA) probes is synthesized on-chip. Labelled samples DNA are then added and hybridized so the identity/abundance of complementary sequences are determined. This method is "historically" called DNA chips and it was originally developed at Affymetrix Inc. [LS02].

Microarrays have significant advantages because they may contain a very large number of genes and are small. Therefore, they are useful when one wants to survey a large number of genes quickly or when the sample to be studied is small. The microarray (DNA chip) technology is having a significant impact on genomics study. Many fields, including drug discovery and toxicological research, will certainly benefit from the use of DNA microarray technology, which will be more thoroughly discussed in the next section.

### 2.3.2 Applications of Microarray Technology

Microarray technology may be used in a wide range of applications.

#### **Gene discovery**

Genomic and gene expression microarray experiments can be used to identify new genes involved in a pathway. Potential drug targets or expression markers can then be used in a predictive or diagnostic fashion.

#### **Disease diagnosis**

Micoarrays are very valuable for understanding biological processes and understanding and treating human diseases. For example, we can find gene expression (mRVA) markers

after analyzing multiple samples obtained from individuals with or without acute leukemia or diffuse large B-cell lymphoma. Based on the markers, these cancers can be classified [MJL01] [GST99] [ALI00] [CA02] [LH02].

**Drug discovery: *Pharmacogenomics***

*Pharmacogenomics* is the hybridization of functional genomics and molecular pharmacology. The goal of pharmacogenomics is to find correlations between therapeutic responses to drugs and the genetic profiles of patients. We can find the target of drugs by comparing the expression profile of a drug-treated cell with the profiles of cells in which single genes have been individually inactivated. For example, microarrays were used to identify a drug ‘mechanism’ by utilizing the Rosetta data set [HUG00].

**Toxicological research: *Toxicogenomics***

*Toxicogenomics* is the hybridization of functional genomics and molecular toxicology. The goal of toxicogenomics is to find correlations between toxic responses to toxicants and changes in the genetic profiles of the objects exposed to such toxicants. Through this, we can classify drugs and their modes of action [UF02]. For example, the functional similarity and specificity of different purine analogues have been determined by comparing the genome-wide effects on treated yeast, murine, human cells [GRA98] [MJL01].

• Other existing applications of microarrays include: gene expression under control environment and test condition, developmental time course studies, resequencing, mutation analysis, genotyping, *etc.* Finally, it is anticipated that there will be more applications in the future.

## 2.4 The usage of clustering for microarray analysis

Microarray analysis provides a systematic genome-wide approach to solve the problems already enumerated above. Clustering techniques manifest their crucial power as the first step in microarray analysis.

Clustering algorithms can be applied on gene expression data under various conditions or across different tissue samples to group together genes that have similar functions. For example, Rioult *et al* [RBC03] analyzed expression matrices to identify a *prior* interesting set of co-regulated genes. They proposed a method that can process the transposed matrices by making use of properties of the Galois connections. This technique processes the transposed matrices while computing the sets of genes. It can deal with expression matrices that are dense and have generally only a few lines. They also validated the potential of this framework by looking for the closed sets in two microarray data sets: where one data set concerns the study of human insulin-resistance and the other concerning gene expression during the development of the drosophilae. The results show that this method can efficiently extract patterns from huge gene expression databases [RBC03].

Gene clustering also has become the first step to uncover the regulatory elements in transcriptional regulatory networks [CPM02][GMC02].

Cohen *et al.* [CPM02] applied microarray analysis to Yap1p and Yap2p Transcriptional Networks and obtained the discrimination between Paralogs. The research shows that DNA microarray can distinguish the functions of two closely related homologues from the yeast *Saccharomyces cerevisiae*, Yap1p and Yap2p, using microarray clustering. Focusing on expression clusters that are over represented for Yap binding sites helps in distinguishing direct versus indirect effects on transcription caused



by transcription factors. Their approach allows the identification of clusters with unexpected expression patterns. It is also easily scalable to larger genomes and larger protein families.

Another example is the identification of unstable transcripts in *Arabidopsis* by cDNA microarray analysis [GMC02]. It is found that Rapid decay is associated with a group of touch- and specific clock-controlled genes. It is pointed out that genes with unstable transcripts often encode proteins that play important regulatory roles. In this research, cDNA microarray analysis was applied to identify and characterize genes with unstable transcripts in *Arabidopsis thaliana* (ATGUTs). Results show that mRNA instability is of high significance and is associated with specific genes controlled by the circadian clock. For the analysis of gene expression data across multiple experiments the CLUSTER and TREEVIEW software were used.

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. Scherf [SRW00] 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.* [GST99] 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 [GST99]. It has been indicated that clustering samples can be used to identify fundamental subtypes of any cancer [SRW00].

To decide which and how many genes should be selected for further studies is an important issue in the microarray data analysis area. Clustering for microarray analysis can also be used for gene selection [LG03].

Clustering analysis can also be used to find direct gene-sample correlations. BiCluster [CB02] 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 [CB02] [MD01]. Domany [DE02] 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.

## **2.5 Genetic algorithms**

The famous naturalist Charles Darwin defined Natural Selection or Survival of the Fittest as the preservation of favourable individual differences and variations, and the destruction of those that are injurious. In nature, individuals must adapt to their environment to survive. This process is called evolution, through this procedure the fittest genes survive and are transmitted to their descendants during the replica process and sexual recombination process, which is called crossover.

In the late 60s, Holland applied natural selection to machine learning using a technique that was later named genetic algorithms. In 1989, Goldberg provided a solid scientific basis for this area, and cited some successful applications of the genetic algorithm. In recent years there is more software and literature devoted to this subject.

Genetic algorithms are modeled after mechanisms of natural selection. Each optimization parameter ( $x_i$ ) is encoded by a gene using a real number or a string of bits. The corresponding genes for all parameters  $x_1, \dots, x_n$  form a chromosome, which describes an individual design solution, and a set of chromosomes represent several individual design solutions, with those fittest ones being selected to reproduce. Crossover is performed to combine genes from different parents to produce children. The children are inserted into the population and the procedure starts over again. This procedure creates an artificial Darwinian environment. Cross-over may be classified as single point or multiple points. To illustrate the process consider the following chromosomes where the genes are encoded as binary bits.

101101110111 and 110011101011; a single point cross-over is specified as 6 to divide each chromosome into two parts such that the first 6 genes are in one part and the rest are in the other part. Then the first part of the first chromosome is combined into a new chromosome with the second part of the second chromosome and the first part of the second chromosome is combined with the second part of the first chromosome into another new chromosome to get the two chromosomes: 110011110111 and 101101101011.

Mutation changes the value of a single bit and helps in preventing being stuck on a local optimal. The traditional mutation operator randomly chooses and flips a bit, changes the bit from 1 to 0 or 0 to 1.

After applying the genetic operators such as crossover and mutation, the "offspring" generated will include solutions better than the purely random original ones. The best offspring will be added to the population while inferior ones will be eliminated. By

repeating this process, repeated improvements will occur in the population, until we finally get the same result repeated for certain generations.

A genetic algorithm for a particular problem must have the following five components [CCA98]:

1. Chromosome representation.
2. The way to create an initial chromosome population.
3. Suitable fitness function.
4. Genetic operators that alter the composition of children (crossover, mutation, reproduction *etc.*)
5. Values for various genetic algorithm parameters (population size, probabilities of applying genetic operators, *etc.*).

## 2.6 Multi-objective GA's

A problem is said to be multi-objective if it involves simultaneous optimization of multiple goals. Usually there is no single solution for which all objectives are optimal. For example, a solution may be optimal regarding one objective but inferior regarding another objective, so the design goals are competing and there will be some trade-offs.

In general multi-objective problems do not have a unique solution and the solution of a multi-objective problem is a set of solutions, such that there are no other solutions that are superior in comparison to all other objectives. The solution set is called Pareto-optimal set or non-dominated set. This concept was formulated by Pareto in 1896 [VP1896], and constitutes the origin of research in multi-objective optimization.

Assume a multi-objective problem has  $k$  objectives. Assume this is a minimization problem, for solution set  $\mathbf{x} = (x_1, x_2, \dots, x_k)$  and another solution set  $\mathbf{y} = (y_1, y_2, \dots, y_k)$ , if

for all  $i$ ,  $x_i \leq y_i$  and there exist  $i$  such that  $x_i < y_i$ . Then solution  $x$  is said to dominate solution  $y$ . If there is no other solution  $y'$  such that  $y'$  dominates  $x$ , then solution  $x$  is a member of the Pareto-set is said to be *non-dominated*. Thus, the multi-objective problem can be defined as finding solutions that are non-dominated [ND99].

The minima in the Pareto sense are going to be in the boundary of the design region called the Pareto front. For example, considering two objective functions, one is to get the minimum number of clusters and the other one is to get the smallest fitness value. Assuming a data set:  $\{(2, 32.2), (3, 30.4), (4, 29.7), (5, 29.0), (2, 31.5), (4, 28.8)\}$ , the first value is the number of clusters and the second value is the fitness value. The subset  $\{(2, 31.5), (3, 30.4), (4, 28.8)\}$  is the Pareto front, because there is no data point in the whole data set that has both less cluster number and smaller fitness value. That is, in the Pareto set, there is no data point dominated by other data point. Figure 2.2 shows an example of Pareto Front.

### 2.6.1 Multi-objective optimization

In general multi-objective design problem it can be expressed by Equations (2.2).

$$\begin{aligned} \text{Min } F(x) &= (f_1(x), f_2(x), \dots, f_k(x))^T \\ \text{s.t. } x &\in S \\ x &= (x_1, x_2, \dots, x_n)^T \end{aligned} \tag{2.2}$$

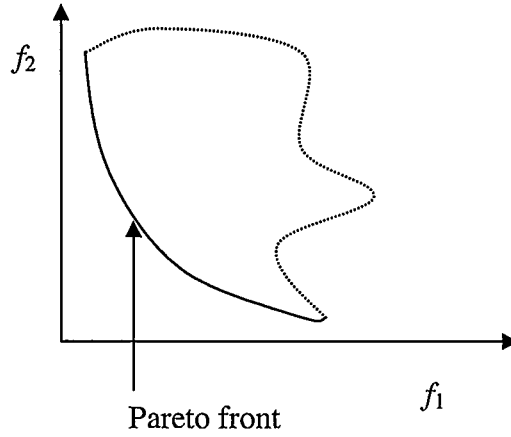


Figure 2.2 An example of a problem with two objective functions.  
(The Pareto front is marked with a continuous line that faces the two axes.)

where  $f_1(\mathbf{x})$ ,  $f_2(\mathbf{x})$ , ...,  $f_k(\mathbf{x})$  are the  $k$  objective functions,  $(x_1, x_2, \dots, x_n)$  are the  $n$  optimization parameters, and  $S \in R^n$  is the solution or parameter space.

The obtainable objective vectors,  $\{F(\mathbf{x}) | \mathbf{x} \in S\}$ , are usually referred to as the attribute or objective space. It can be:

1. The Pareto (non-dominated) set which consists of solutions that are not dominated by any other solutions.
2. A dominated set, if there exist a solution that dominates it.
3. The space in  $R^k$  formed by the objective vectors of Pareto optimal solutions is known as the Pareto optimal front.

In a general multi-objective GA process, initial population is randomly generalized. Then with crossover, mutation, and selection (for multi-objective GA is Pareto selection) which are traditional genetic processes, solution chromosomes evolves to more optimal ones, which means both objective function values are getting better. The process can be showed in the following figure:

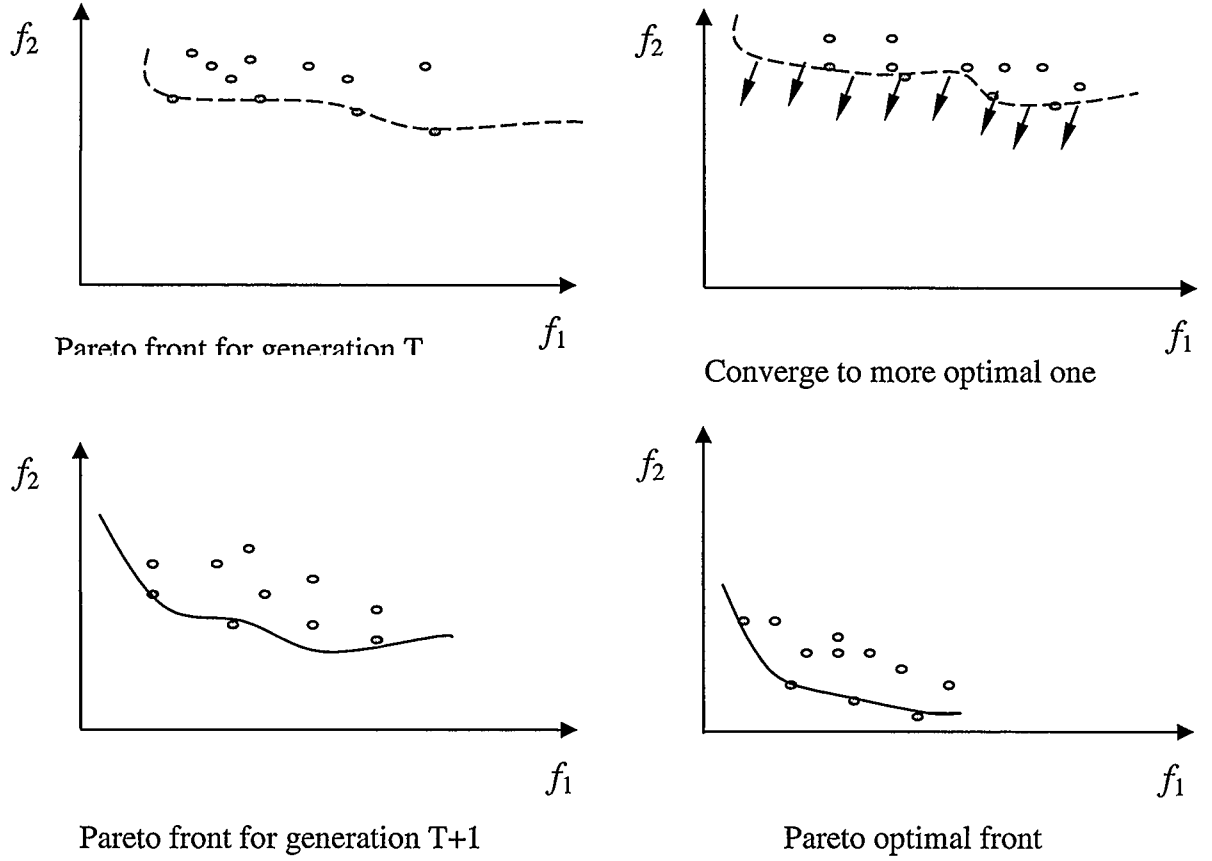


Figure 2.3 Illustration of Pareto front convergence process

In Figure 2.3, blue points denote Pareto set data points for the current population. It is shown that, along with the genetic process, the Pareto front converges to the optimal one and remains stable. For example, considering two objective functions:  $f_1 = \min(a)$ ,  $f_2 = \min(b)$ , for generation one we have solution set  $\{(2, 8), (2, 11), (3, 7.7), (3, 13), (4, 6.5), (5, 5)\}$ . For each solution, the first value is  $a$  and the second one is  $b$ , then the Pareto set is  $\{(2, 8), (3, 7.7), (4, 6.5), (5, 5)\}$ . After crossover and mutation, solution chromosomes evolves to  $\{(2, 7), (3, 7.7), (4, 6.5), (5, 4.5), (2, 10), (3, 9)\}$ , and the Pareto front for generation two is  $\{(2, 7), (4, 6.5), (5, 4.5)\}$ , .... The evolution will keep on going until the Pareto front set becomes stable. The final Pareto front is  $\{(2, 3.2), (3, 2), (4, 1.8), (5,$

0.9)}. It can be seen that for each  $a$  value we get a smaller  $b$  value in the final Pareto front.

## 2.6.2 Multi-objective Genetic Algorithms

In the past decade, several evolutionary approaches to multi-objective problems have been introduced.

One commonly used approach is the usage of weighting coefficients and penalty functions to combine the objective functions into a single objective. This transformation allows us to use a simple single-objective genetic algorithm for a single solution. However, it may not fulfill all designer's needs so the solution may not even be non-dominated. It also requires search space knowledge to set the weights, which is often unavailable. Therefore, this method is not always applicable and efficient.

Another approach is to use genetic algorithm to locate Pareto-optimal solutions. Some researchers show that it is more effective [SCH85]. Evolutionary algorithm approaches are particularly suitable to solve multi-objective problems because they deal simultaneously with a set of possible solutions and this allows us to find several members of the Pareto optimal set in a single run of the algorithm. Another reason is that evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front. They can easily deal with discontinuous or concave Pareto fronts.

Multi-objective genetic algorithm approaches include Vector Evaluated Genetic Algorithm (VEGA), Multi-Objective Genetic Algorithm (MOGA), Niche Pareto Genetic Algorithm (NPGA), and Non-dominated Sorting Genetic Algorithm (NSGA).



#### 2.6.2.1 Vector Evaluated Genetic Algorithm (VEGA)

Schaffer [SCH85] developed the method called VEGA (Vector Evaluated Genetic Algorithm) that includes multiple objective functions. The only difference with usual genetic algorithm is the way that the selection is carried out for recombination. In this method, on each generation, the population groups in a certain number of subpopulations according to each objective function. These are then shuffled together to obtain a population so that conventional crossover and mutation are performed.

This approach can work properly in simple multi-objective optimization problems. It is easy to implement and it is efficient. However, it cannot generate Pareto optimal front when it is concave.

#### 2.6.2.2 Multi-Objective Genetic Algorithm (MOGA)

Multi-Objective Genetic Algorithm (MOGA) was proposed by Fonseca and Fleming in 1993 [CFP93]. In this method, an individual is ranked according to the number of chromosomes in the current population by which it is dominated. All non-dominated individuals are assigned rank 1 and dominated individuals are penalized according to the population density of the corresponding region of the trade-off surface.

Fitness values are assigned according to the following process: sort population according to rank; assign fitness to individuals by interpolating from the best (rank 1) to the worst (rank  $n$ ); average the fitness values of individuals with the same rank. The authors use a niche-formation method to distribute the population over the Pareto-optimal region, which can prevent premature convergence.

MOGA normally outperforms all of its contemporary competitors. It is efficient and relatively easy to implement but its performance is highly dependent on an appropriate selection of the sharing factor. Other Pareto techniques have the same problem.

#### 2.6.2.3 Niched Pareto Genetic Algorithm (NPGA)

Niched Pareto Genetic Algorithm was proposed by Horn *et al.* in 1993 [HN93]. It uses a tournament selection scheme based on Pareto dominance. Two individuals randomly selected are compared against a subset from the entire population. When both competitors are either dominated or non-dominated, the result of the tournament is decided through fitness sharing in the objective domain.

It seems that this method has good overall performance. It is efficient because it does not apply Pareto ranking to the entire population and it is easy to implement. In addition, it requires another parameter (tournament size) in addition to a sharing factor.

#### 2.6.2.4 Non-dominated Sorting Genetic Algorithm (NSGA)

Proposed by Srinivas and Deb in 1994 [SND94], NSGA is based on several layers of classifications of the individuals. In this method, non-dominated individuals get a certain dummy fitness value, and are then removed from the population. The process is repeated until the entire population has been classified. To maintain the diversity of the population, classified individuals are shared (in decision variable space) with their dummy fitness values. This method is sensitive to the value of the sharing factor and it is relatively easy to implement.

## 2.7 Related work

Existing literature shows that increasing attention is devoted to the development of new clustering techniques. As mentioned in the previous sections, existing clustering techniques which are mostly used for gene expression data clustering can be classified into traditional clustering algorithms including hierarchical clustering [ND99], partitioning [CCA98], and recently emerging clustering techniques such as graph-based [MD01] and model-based [SCD03] [WK00] approaches.

Hierarchical clustering has several advantages: it is robust with respect to input parameters, less influenced by cluster shapes, less sensitive to largely differing point densities of clusters, and it can represent nested clusters. However it suffers from different aspects as stated by statisticians, including robustness, non-uniqueness, and inverse interpretation of the hierarchy [PT99]. In addition, its tree structure is prone to errors, since there is multi-ways of expressing the similarity. This gets worse as the data size increase [MA95]. Once a gene is assigned to a cluster, there is no possibility of assigning it to another cluster to see whether there are better results. On the basis of traditional hierarchical clustering method, Segal and Keller [SK02] propose probabilistic abstraction hierarchies (PAH), where each class is associated with a probabilistic generative model for the data in the class. This method improved the performance of traditional hierarchical clustering by handling the drawbacks mentioned above. It is more robust and less sensitive to noise in data.

Partitioning algorithms create a “flat” decomposition of a data set. Examples of partitioning algorithms are  $K$ -means, SOM, and DSOM. The  $K$ -means algorithm is widely used in microarray data analysis. The shortcoming of this method is that it finds the local optimum but may miss the global one. This clustering process is not a stable one

because of the initial phase so that at every run it is more probable to obtain different clustering results.

Self Organizing Maps (SOM) is popular in vector quantization. It uses an incremental approach – points (patterns) are processed one-by-one. It allows mapping centroids into 2D plane that provides a straightforward visualization. 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 the prior information about the distribution of the data is not available.

The model-based approach assumes that data are generated by a mixture of finite number of probability distributions. There is a tradeoff between the complexity of the probability model and the number of clusters. For instance, if a complex probability model is used, a small number of clusters may suffice, while if a simple model is used, a larger number of clusters may be needed to fit all the data appropriately. Examples of model-based approach are Bayesian method and mixture model-based algorithm (EMMIX-GENE). The Bayesian method has the advantage that it can identify the number of distinct clusters but it has the disadvantage of relying on the assumption that the modeled time series are stationary [YN01]. Mixture model-based algorithm (EMMIX-GENE) clustering results show that it sometimes has errors [MM03].

Graph-based methods also have some shortcomings. Although the number of clusters is not given, there is a pre-specified threshold used for the clustering. After the convergence, each gene moves to the cluster with the highest average similarity. This is a very expensive cleaning step.

The method proposed in this thesis 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 fixed number of clusters *a priori*, this thesis is keen on giving a range for the number of clusters parameter and finding a set of 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 the traditional multi-objective algorithms that scalarize the objectives by assigning subjective weights to each function. Hence, we donot need consider weights in the system. In addition, using a genetic algorithm with recombination and mutation, we can find the global optimum solution using the appropriate system run parameters.

In summary, the method presented and analyzed in this thesis is unique in presenting the set of solutions in the Pareto optimal front and analyze their validity to select the most appropriate 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 proposed approach.

## Chapter Three

### The Approach

A new clustering approach named Multi-Objective Genetic *K*-means algorithm (MOKGA) is proposed here. It is a general-purpose approach for clustering other datasets as well, after modifying the fitness functions and changing the proximity values as distance or non-decreasing similarity function according to the requirements of datasets to be clustered. It has been developed on the basis of the Fast Genetic *K*-means Algorithm (FGKA) [YLU04] and the Niche Pareto Genetic Algorithm [HNG94].

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 under six cluster validity techniques proposed in the literature, namely Silhouette, *C* index, Dunn's index, SD index, DB index, S\_Dbw index.

This chapter 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 talks about the fitness evaluation and selection in MOKGA. Section 3.4 discusses the mutation and cross over operations. The implementation details are described in Section 3.5.

### 3.1 The objectives

During the clustering process, two objective functions are defined: minimizing the partitioning error and minimizing the number of clusters.

To partition the  $N$  pattern points into  $K$  clusters, one goal is to minimize the Total Within-Cluster Variation ( $TWCV$ ), which is specified as:

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

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

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

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

$$F = \min (\text{number of clusters}) \quad (3.3)$$

After running the algorithm, the aim is obtaining the first Pareto optimal front having the best partition 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  $\{1, 2, \dots, K\}$  and represents a pattern. The value indicates the cluster to which the corresponding pattern 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 different values from the interval  $[1 \dots k]$ .

### 3.3 Fitness evaluation and selection

The fitness value for each chromosome is a Total Within-Cluster Variation (*TWCV*) value. The calculation of the value has been discussed in Section 3.1. In this thesis, the Niche Pareto tournament selection scheme is used for the selection 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. 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 Niche Pareto Tournament Selection. Where if neither of the two is dominated by the comparison set then they will use sharing to choose a winner [HNG94], which is not necessarily in this system.

As an example, Figure 3.1 shows two data points:  $P_1$  and  $P_2$ , and a comparison data set represented by a curved line.  $P_1$  is dominated by the comparison set, because it has a bigger *TWCV* value or a bigger number of clusters in comparison with every data point in the comparison set, so it will be deleted from the population;  $P_2$  is not dominated, because it has a smaller *TWCV* value in comparison with the data point with the same cluster number in the comparison set, so it will be kept for the next step.



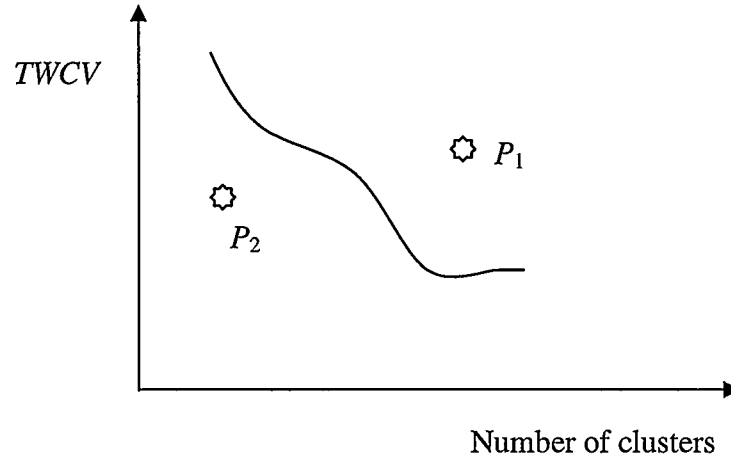


Figure 3.1. An example of Pareto tournament selection

### 3.4 Crossover and mutation

#### Crossover operator:

One-point crossover operator is applied on two randomly chosen chromosomes. The crossover operation is carried out on the population with the crossover  $p_c$  (crossover rate). After the crossover, assigned cluster numbers for each gene are renumbered beginning from  $a_1$  to  $a_n$ . For example, if two chromosomes having 3 clusters and 5 clusters, respectively, they 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, which are then 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 some initial experiments demonstrated that one-point cross-over produces better fitness values than multi-point attempts.

### **Mutation operator:**

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  $n = 1, \dots, N$  simultaneously.  $a_n'$  is a cluster number randomly selected from  $\{1, \dots, K\}$  with the probability distribution  $\{p_1, p_2, \dots, p_K\}$  defined using the following formula:

$$P_i = \frac{1.5 * d_{\max}(\overline{X}_n) - d(\overline{X}_n, \overline{C}_k)}{\sum_{k=1}^K (1.5 * d_{\max}(\overline{X}_n) - d(\overline{X}_n, \overline{C}_k))} \quad (3.4)$$

where  $i = (1, 2, \dots, k)$  and  $d(X_n, C_k)$  denotes the 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 a mutating gene is 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  $k$ -th cluster  $G_k$ .

### **3.5 Implementation details**

The gene expression data clustering system proposed in this thesis consists of two components: the Multi-Objective Genetic K-means Algorithm (MOKGA) cluster and the cluster validity component. The implementation details are described in the following sections.

### 3.5.1 Multi-Objective Genetic $K$ -means Algorithm (MOKGA)

As presented in the flowchart shown in Figure 3.2, 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 in [PS87] 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.

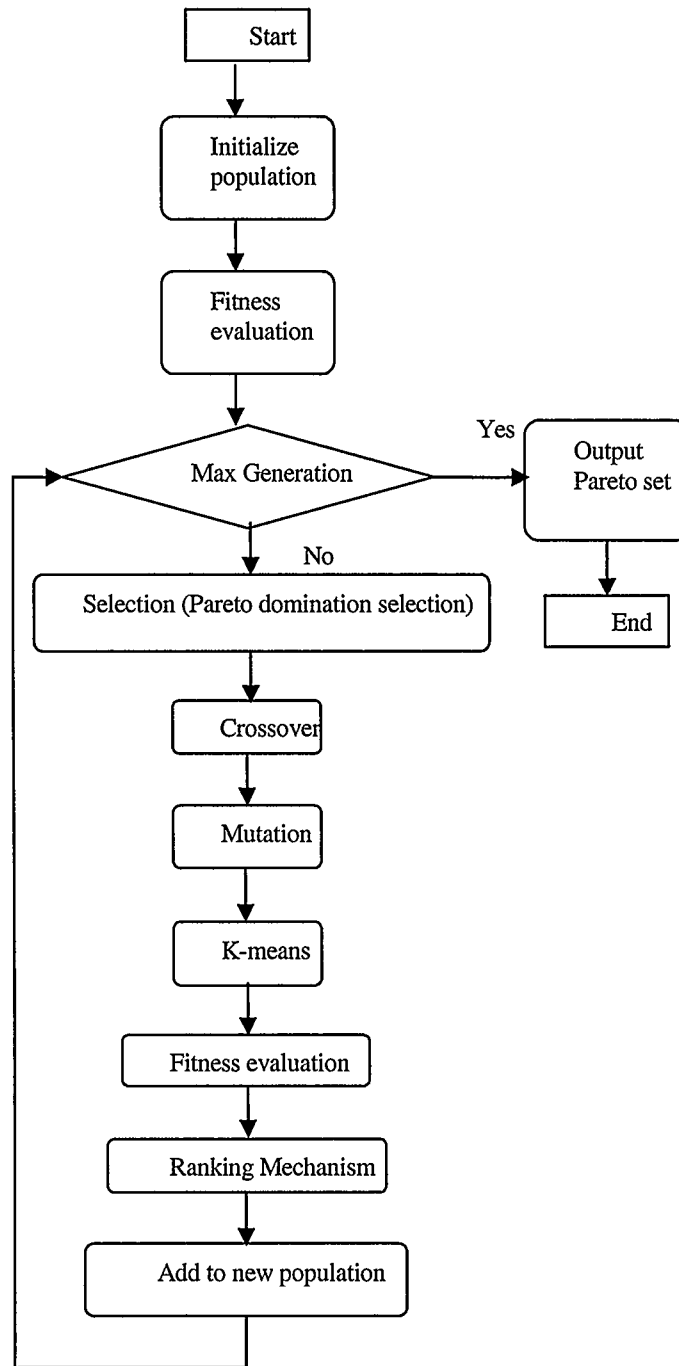


Figure 3.2. Flow chart: the process of the Multi-Objective Genetic K-means Algorithm

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, the k-means operator [YLU04] are then performed 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 in [EZI99] is applied to satisfy the elitism and diversity preservation. This halves the number of individuals.

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 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 cluster values 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 is:

**Function crossover****Begin**

```
/* Randomly chose the two chromosomes*/  
  
Chromosome_1 = rand() % biggest chromosome index  
Chromosome_2 = rand() % biggest chromosome index  
  
/* Randomly chose 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 probability distribution, as discussed in Section 3.4. The pseudo code is as following:

**Function mutation****Input:** population  $P$  ( $S_1, S_2, \dots, S_J$ ), Mutation probability MP**Output:** population  $P'$  ( $S'_1, S'_2, \dots, S'_J$ )**Begin**  **For**  $j=0$  to  $J$  **do** /\* for each solution  $S_j$  in population  $P$ \*/     $SD=0$ ; /\*summation of distribution\*/     $\overline{C_1} \dots \overline{C_k} = \text{CalCentroids}(S_j)$  /\* calculate the centre point for each cluster\*/    **For**  $n=1$  to  $N$  **do** /\*for each data point in  $S_j$ \*/      **If**  $\text{rand}() < MP$  **then**         $d\_max=0.00$ ;        **For**  $k=1$  to  $K$  /\* for each cluster \*/           $d_k = \text{calEuclideanDistance}(\overline{X_n}, \overline{C_k})$  /\* distance from data to cluster

centre\*/

 $d\_max = \max(d\_max, d_k)$            $SD = SD + (1.5 * d\_max(\overline{X_n}) - d(\overline{X_n}, \overline{C_k}))$         **End For**         $p_1 = (1.5 * d\_max - d_1) / SD$  /\* Mutation probability for cluster 1\*/        **For**  $k=2$  to  $K$ 

/\* Mutation probability for cluster 2~ CLUSTER\*/

 $p_k = (1.5 * d\_max - d_k) / SD + p_{k-1}$ ;        **End for**         $S'_j.a'_n$  = a cluster number, randomly chose according to the distribution  $p_1,$  $p_2, \dots, p_k$       **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 cluster centre for each cluster and re-assigns each gene to the closest cluster to each instance in the gene. Hence,  $K$ -means operator is used to speed up the convergence process by replacing  $a_n$  by  $a'_n$ , for  $n=1, \dots, N$  simultaneously, where  $a'_n$  is the closest to object  $X_n$  in Euclidean distance. The pseudo code for  $K$ -means operator is given in the following:



**Function K-Means operator****Input:** population  $P (S_1, S_2, \dots, S_J)$ **Output:** population  $P' (S'_1, S'_2, \dots, S'_J)$ **Begin**    **For**  $j=1$  to  $J$  **do**      /\* each solution in a population  $P$  \*/         $\overline{C}_1 \dots \overline{C}_k = \text{CalCentroids}(S_j)$    /\* calculate the centre point for each cluster \*/        **For**  $n=1$  to  $N$  **do**   /\* each data point in a solution \*/             $d_{\min} = \text{MAX\_NUMBER}$             **For**  $k=1$  to  $K$  **do**      /\*  $K$  is maximum cluster number \*/                 $d_k = \text{calEuclideanDistance}(\overline{X}_n, \overline{C}_k)$    /\* calculate the Euclidean distance from the data point to each cluster centre \*/                **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 operators have been applied, twice the number of individuals remains. 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 in [EZI99] is employed. Thus, the individuals obtained after crossover, mutation, and the  $K$ -means operator are ranked, we pick among them the best individuals to place in the population for the next generation.

The 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 the  $2l$  set and place it in the

population to run in the next generation. In the remaining sets we get the first Pareto-optimal front and 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. We then sum the normalized value 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. The individuals are placed in 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 that we have 6 in the current population, we take 2 chromosomes that have the biggest distance to their neighbours using the ranking method that we 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 the next generation is performed.

### 3.5.2 Cluster Validity

Concerning the employed 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. The system analyzes the clustering results found under six of the cluster validity techniques proposed in the literature, namely Silhouette,  $C$  index, Dunn's index, SD index, DB index, and S\_Dbw index. The calculation of the indices is described in the following sections.

#### 3.5.2.1 The SD validity index

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:

$$Scatt(n_c) = \frac{1}{n_c} \sum_{i=1}^{n_c} \frac{\|\sigma(v_i)\|}{\|\sigma(x)\|} \quad (3.5)$$

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:

$$Dis(n_c) = \frac{D_{\max}}{D_{\min}} \sum_{k=1}^{n_c} \left( \sum_{i=1}^{n_c} \|v_k - v_z\| \right)^{-1} \quad (3.6)$$

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

Finally the SD index is calculated using the following equation:

$$SD(n_c) = \alpha Scat(n_c) + Dis(n_c) \quad (3.7)$$

where  $\alpha$  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 in to incorporate both terms in a balanced way. The number of clusters that minimizes the index is an optimal value.

### 3.5.2.2 $S\_Dbw$ validity index

$S\_Dbw$  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:

$$Dens\_bw(n_c) = \frac{1}{n_c(n_c - 1)} \sum_{i=1}^{n_c} \left( \sum_{\substack{j=1 \\ i \neq j}}^{n_c} \frac{density(u_{ij})}{\max\{density(v_i), density(v_j)\}} \right) \quad (3.8)$$

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:

$$density(u) = \sum_{l=1}^{n_{ij}} f(x_l, u) \quad (3.9)$$

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:

$$f(x, u) = \begin{cases} 0, & \text{if } d(x, u) > stdev \\ 1, & \text{otherwise} \end{cases} \quad (3.10)$$

where  $stdev$  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 already been defined in the  $SD$  index part.

Finally, the  $S\_Dbw$  is calculated using the following equation:

$$S\_Dbw(n_c) = Scat(n_c) + Dens\_bw(n_c) \quad (3.11)$$

the definition of  $S\_Dbw$  considers both compactness and separation. The number of clusters that minimizes the index is an optimal value.

### 3.5.2.3 Dunn's Validity Index

The Dunn index is calculated using the following equation [SMW03]:

$$D_{n_c} = \min_{i=1, \dots, n_c} \left\{ \min_{j=i+1, \dots, n_c} \left[ \frac{\frac{1}{|C_i| |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y)}{\max_{k=1, \dots, n_c}^2 \left( \frac{\sum_{x \in C_k} d(x, c_k)}{|C_k|} \right)} \right] \right\} \quad (3.12)$$

where  $c_i$  represents the  $i$ -cluster of a certain partition,  $d(x,y)$  is the distance between data point  $x$  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.

#### 3.5.2.4 Davies-Bouldin(DB) Validity Index

The DB index is calculated using the following equation:

$$DB = \frac{1}{n} \sum_{i=1}^n \max_{i \neq j} \left\{ \frac{S_n(Q_i) + S_n(Q_j)}{S(Q_i, Q_j)} \right\} \quad (3.13)$$

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.

#### 3.5.2.5 Silhouette Validation Method

The following formula is used to calculate the Silhouette index:

$$S(i) = \frac{(b(i) - a(i))}{\max\{a(i), b(i)\}} \quad (3.14)$$

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 the clusters.

#### 3.5.2.6 C index

This index is defined as follows:

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}} \quad (3.15)$$

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.

## **Chapter Four**

### **Experimental Results**

This chapter report the experimental results. We start by describing the testing environment. We then present the results obtained for different datasets. For each dataset, the multi-objective GA based clustering approach is employed first to get the Pareto front and then we run six of the well-known validity indices to rank the obtained optimal solutions. The overall ranked results are compared with the singular results reported in the literature for the same datasets.

#### **4.1 The environment used for the experiments**

To evaluate the performance and efficiency of the proposed system consisting of the MOKGA clustering approach and cluster validity analysis, experiments were conducted on computers with the following features:

- Running Windows XP operating system
- Pentium ®4, 2.00 GHz CPU,
- 512 MB RAM

The system was implemented using MS Visual C++. The running platform is Microsoft Visual Studio.NET 2003.



## 4.2 Data sets

Both widely used clustering data mining datasets and microarray data sets are used to test the proposed system. This demonstrates that the system proposed in this thesis works not only for microarray data (gene expression data) application 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.

Three gene expression datasets, Fig2data, cancer (NCI60), and Leukaemia 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 Leukaemia data sets are used for group cell samples.

The description and testing results of each datasets are discussed in the following sections.

### 4.2.1 Ruspini dataset

The Ruspini dataset [RUS70] is popular for illustrating clustering techniques. It has 75 instances with 2 attributes and integer coordinates with  $0 < X < 120$ ,  $0 < Y < 160$ , which might be naturally grouped into 4 sets.

In one study [RUS70], 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 [ROW98]. The same number of clusters was obtained using genetic algorithms using Calinski and Harabasz criterion, Davies and Bouldin cluster validity methods. Values of major parameters used in genetic algorithms in this study are: the number of iterations = 100, the range of exponential

mutation rate: ranges from 10.0 to 0.000001, population size = 200, and crossover probability = 1.00.

The multi-objective genetic algorithm-based approach proposed in this thesis has been run ten times with the following parameters: population size = 100,  $t_{dom}$  (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, 10] was picked for finding the optimal number of clusters.

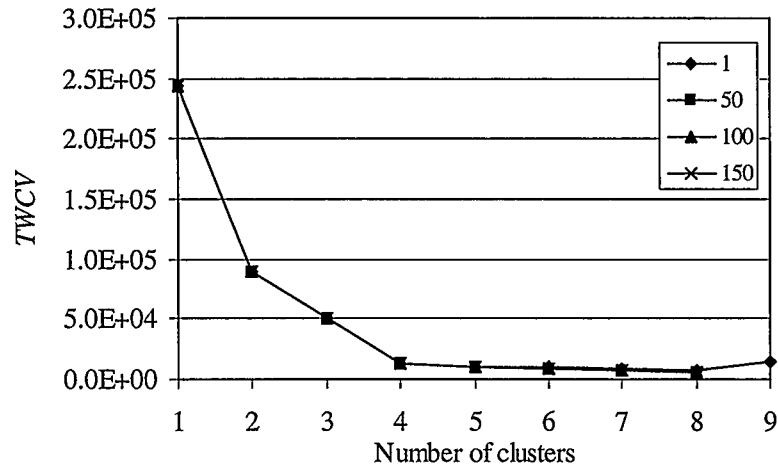


Figure 4.1 Pareto-fronts for Ruspini dataset

Table 4.1 Ruspini Dataset  $TWCV$  for  $k = 8$

Iteration	$TWCV$
1	7718.25
50	6158.25
100	6157.50
150	6149.63

Changes in the Pareto-optimal front by running the algorithm for the Ruspini datasets are displayed in Figure 4.1 for different generations. It demonstrates how the system converges to an optimal Pareto-optimal front. As the actual change in the value of  $TWVC$  is not reflected in Figure 4.1, key  $TWVC$  values are reported in Table 4.1.

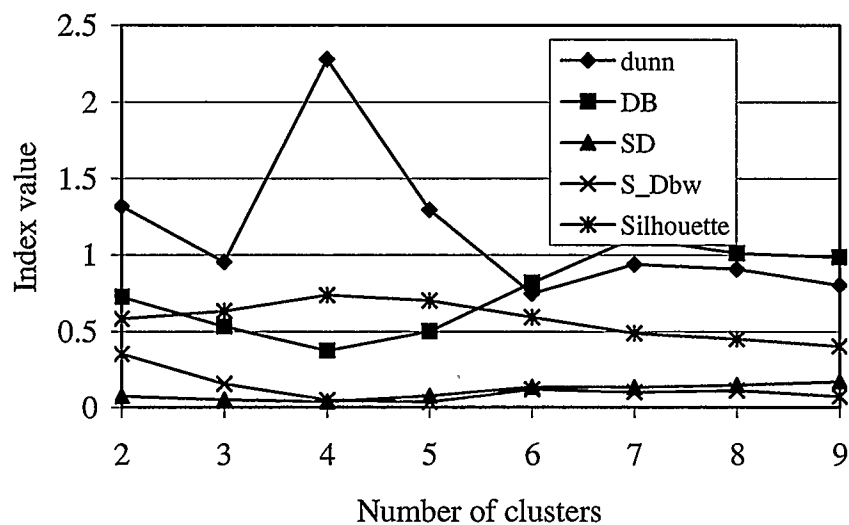


Figure 4.2 Ruspini dataset cluster validity results using Dunn, DB, SD, S\_Dbw and Silhouette indices

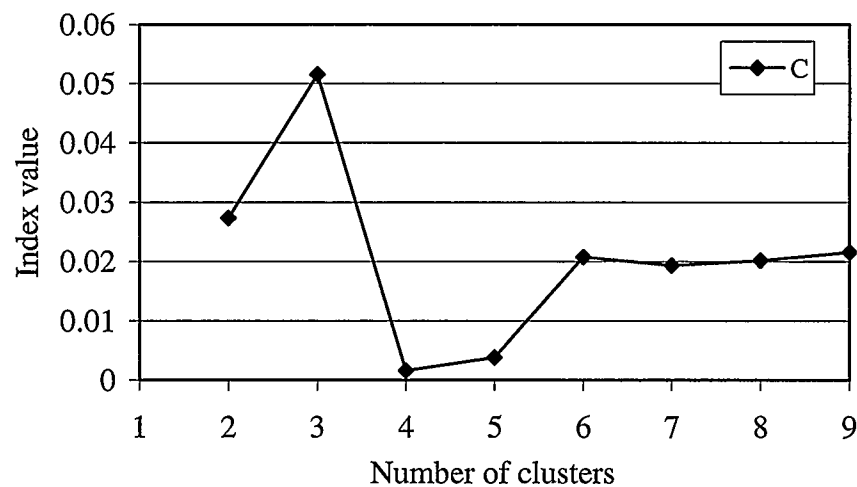


Figure 4.3 Ruspini dataset cluster validity results using C index

In the experiments, not only 4 is in our Pareto optimal front as it can be easily seen from the results plotted in Figure 4.2 and Figure 4.3, but also this value is the best for all the cluster validity analysis indexes (The index values are shown in Appendix A). This finding is consistent with the results obtained before and reported elsewhere [RUS70] [ROW98].

#### 4.2.2 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. That 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 [CL03].

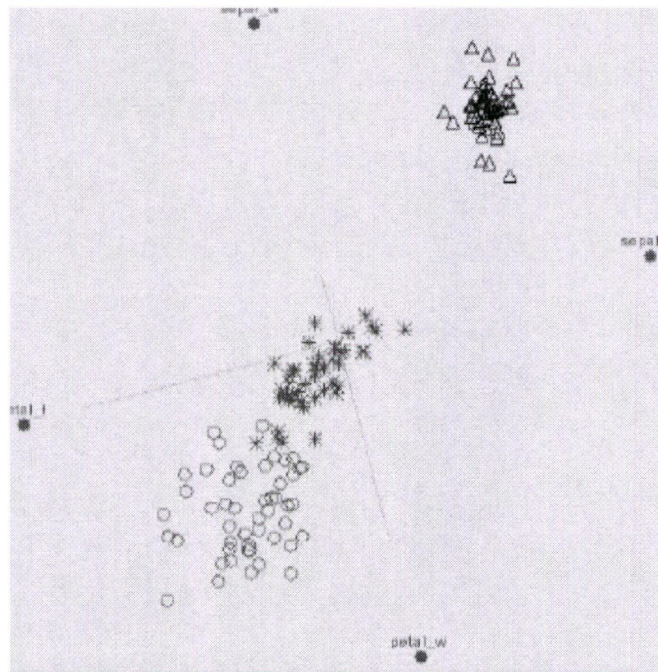


Figure 4.4. The real cluster distribution visualized with the labels from the original Iris dataset: Iris dataset clustering results from [CL03]

Chen and Liu [CL03] applied the Visual Rendering method to Iris dataset. The VISTA system that they used implements a linear and reliable mapping model to visualize  $k$ -dimensional data sets in a 2D star-coordinate space. It allows users to validate and interactively refine the cluster structure based on their visual experience as well 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. Figure 4.4 shows their clustering results for Iris dataset. The same figure also explains why two is the number of clusters in our cluster validity analysis results sometimes have a better index value. Cole also tested the Iris dataset using general genetic algorithms [ROW98]. The values of the main parameters used in the genetic algorithm are: the number of iterations = 1000, the range of exponential mutation rate = from 10.0 to 0.000001, population size = 50, crossover probability = 1.00. For the cluster validity, the optimal numbers of clusters obtained are 3 for Davies Bouldin method and 2 for Calinski and Harabasz method.

The clustering approach proposed in this thesis has been 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 Figure 4.5 for different generations. It demonstrates 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 plotted in Figure 4.5, some key  $TWVC$  values are reported in Table 4.2.

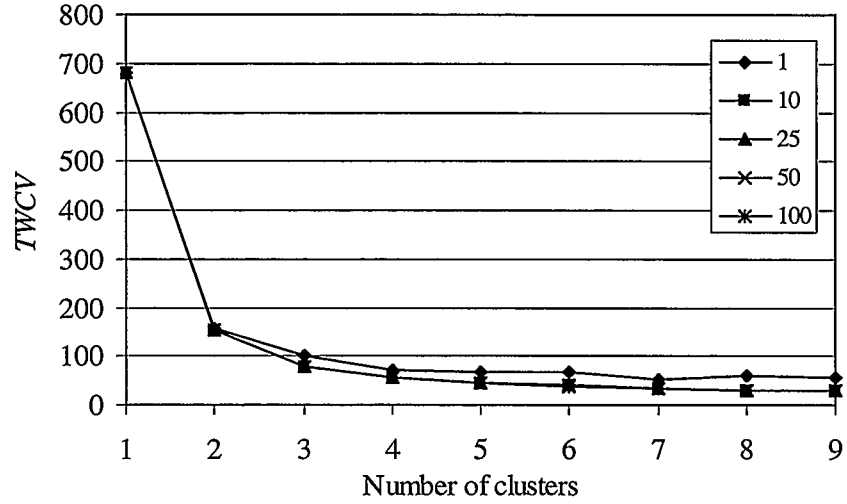


Figure 4.5 Pareto-fronts for IRIS dataset

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

With the Pareto optimal front, the obtained results were tested and analyzed for the Iris dataset using the six indexes mentioned before. The average results of the 10 runs are reported in Figure 4.6, Figure 4.7, Table 4.3, and Table 4.4, respectively.

Finally, the results obtained are compared with the corresponding results reported elsewhere [CL03] [ROW98]. According to [CL03], 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 Figure 4.6 and Figure 4.7). This finding is consistent with the result of the cluster validity DB index published by Rowena [ROW98]. 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 number of clusters taken as 3, while clusters with number of clusters taken as 2 are better separated. The visual clustering results given by Keke and Liu in [CL03] show this difference clearly. C index is likely to be data dependent and the behavior of the index may change when different data structures were used [HBV02].

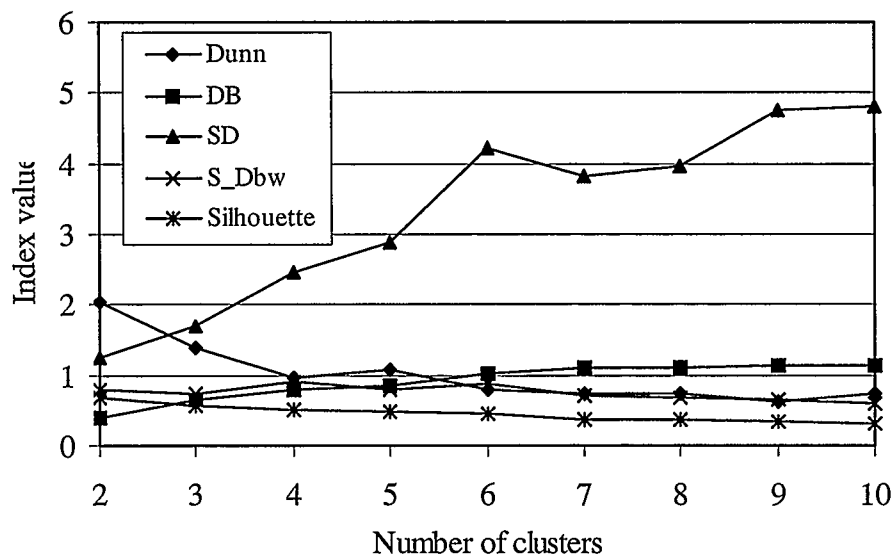


Figure 4.6 Iris dataset cluster validity results using Dunn, DB, SD, S\_Dbw and Silhouette indices

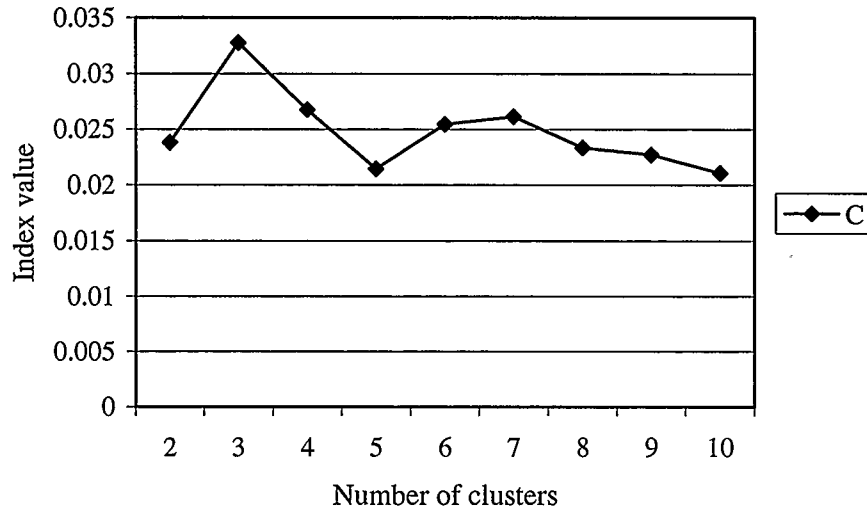


Figure 4.7 Iris dataset cluster validity results using C index

#### 4.2.3 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 expression changed substantially in response to serum. Each gene has 19 expressions ranging from 15 minutes to 24 hours [CL03] [VRI99].

Lu *et al.* [YLU04] 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 proposed genetic algorithm-based approach MOKGA has been applied to 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, which is applied to check if the population stops evolution after 50 generations and if the process needs to be stopped. The range of [1, 25] was picked to find the optimal number of clusters.



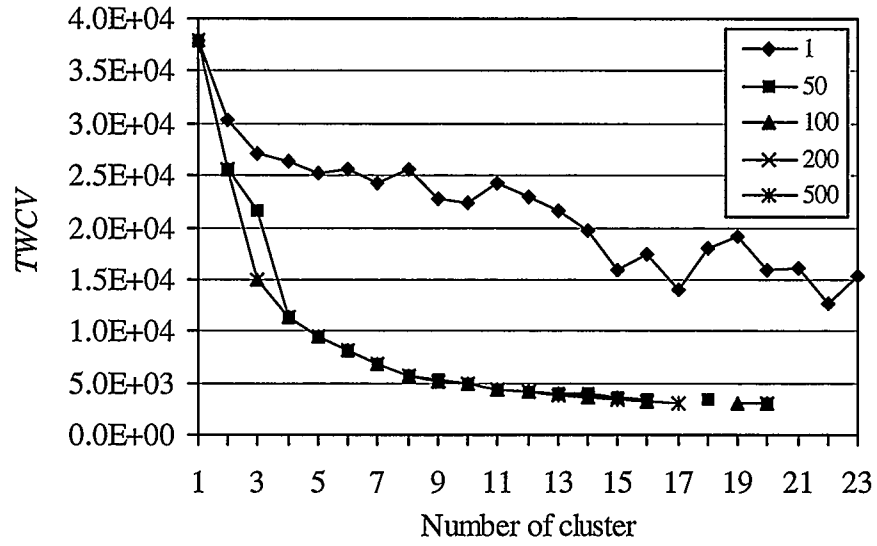


Figure 4.8 Pareto-fronts for Fig2data dataset

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

The corresponding experimental results are demonstrated in Figure 4.8 and Table 4.3. They also show how the system converges to an optimal Pareto front.

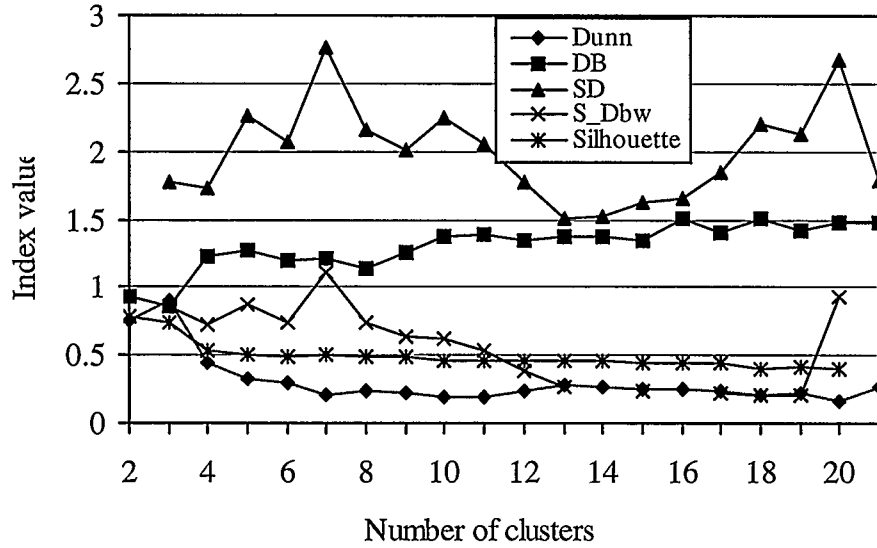


Figure 4.9 Fig2data dataset cluster validity results using Dunn, DB, SD, S\_Dbw and Silhouette indices

Figure 4.9 and Figure 4.10 report validity results and reflect comparisons with the studies described elsewhere [YLU04][VRI99]. The study described by Iyer *et al.* in [VRI99] show that the optimal number of clusters for the Fig2data is 10. Consistently, results in this thesis indicate that it ranks among the best ones for C index, and the number of 10 clusters is among the best for other indices. According to Maria *et al.* [HBV02], SD, S\_Dbw, DB, Silhouette, and Dunn indices cannot handle properly arbitrarily shaped clusters, so they do not always give satisfactory results.

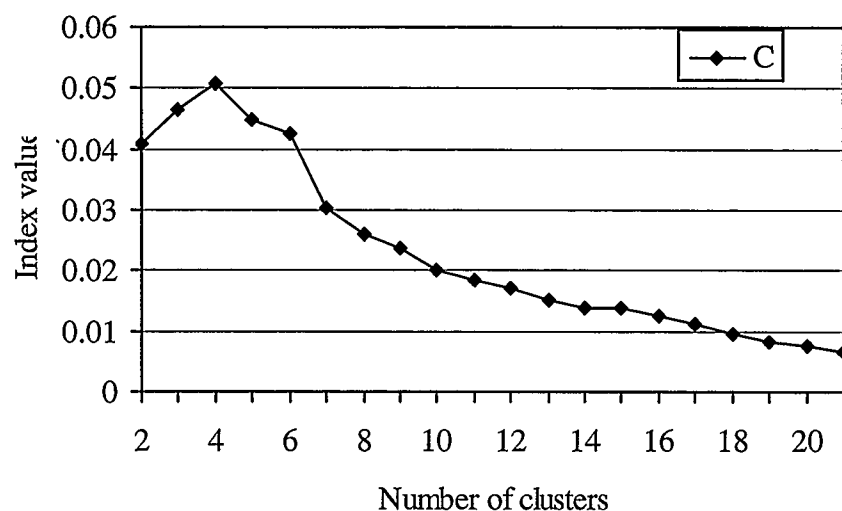


Figure 4.10 Fig2data dataset cluster validity results using C index

#### 4.2.4 Cancer (NCI60) 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, leukaemias and melanomas. Growth inhibition is assessed from changes in total cellular protein after 48 hours 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 [US00]. In the clustering test in this thesis, there is a need to test cell-cell correlations on the basis of drug activity profiles, which are gene expression data available.

The study by Scherf [US00] uses an average-linkage algorithm and a metric based on the growth inhibitory activities of the 1,400 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.

The genetic algorithm-based approach MOKGA proposed in this thesis has been run for the 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.

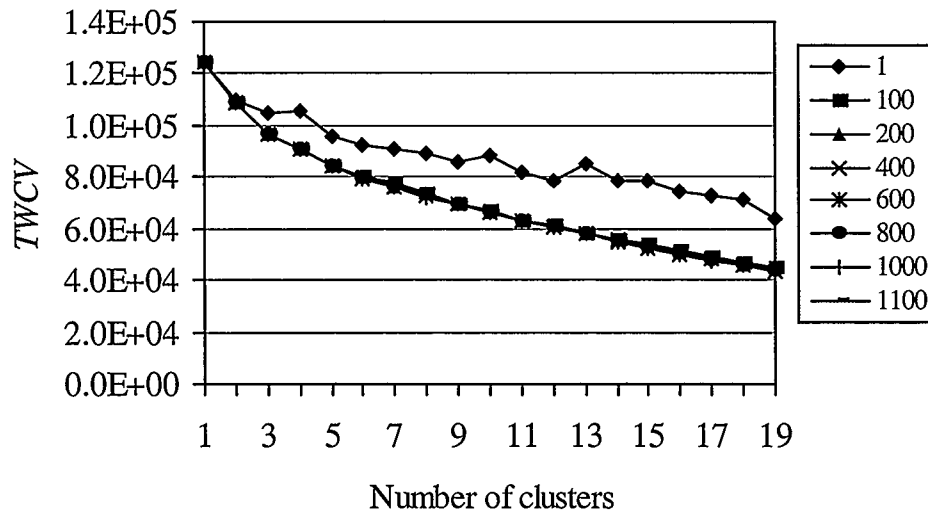


Figure 4.11 Pareto-fronts for Cancer dataset

Table 4.4 Cancer Dataset *TWCV* for  $k = 16$

Iteration	<i>TWCV</i>
1	78435.2
100	53785
200	53210.5
400	52571.8
600	52571.8
800	52398.1
1000	52398.1
1100	52385.3

Changes in the Pareto-optimal front after running the algorithm are displayed in Figure 4.11 and Table 4.4 for different generations. It demonstrates how the system converges to an optimal Pareto-optimal front.

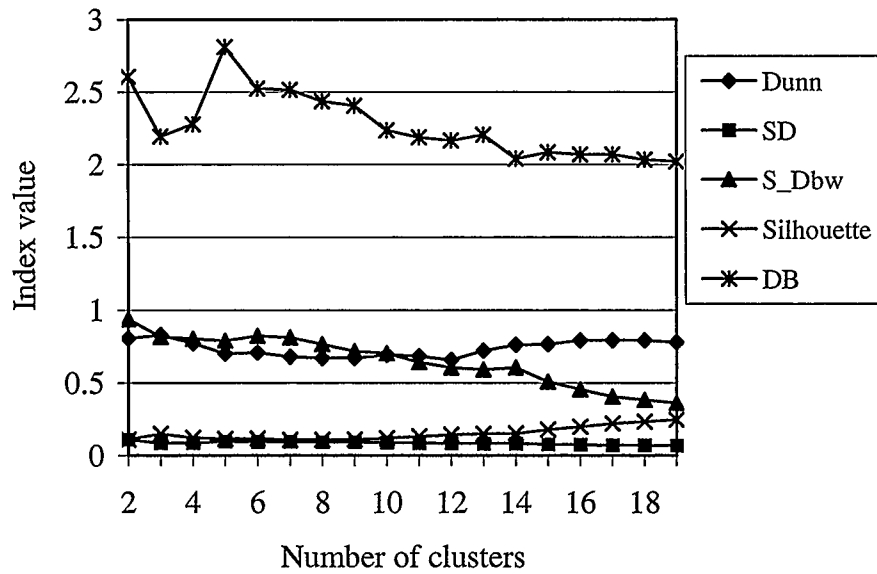


Figure 4.12 Cancer dataset cluster validity results using Dunn, DB, SD, S\_Dbw and Silhouette indices

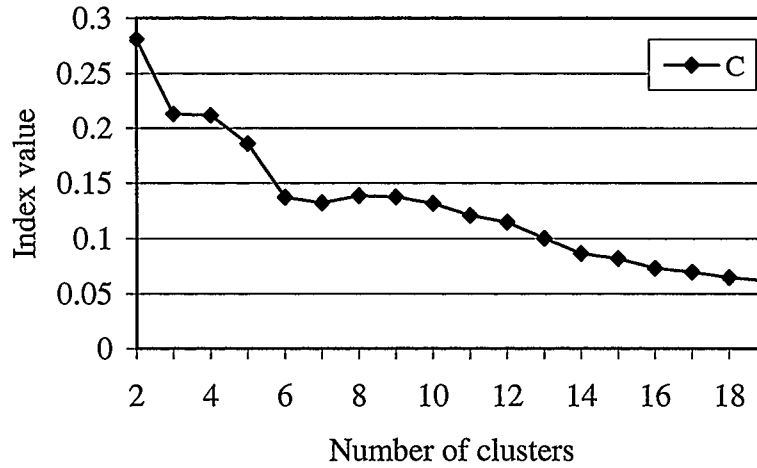


Figure 4.13 Cancer dataset cluster validity results using C index

Figures 4.12 and Figure 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 the fifth for C index. These are consistent with the results reported in [US00]. Some indices values are not good because index values are highly dependent on the shape of the clusters.

#### 4.2.5 Leukaemia dataset

The third microarray dataset used in this thesis is the Leukemia dataset, which has 38 acute leukemia samples and 50 genes. The purposes of the testing include clustering cell samples to groups and finding subclasses in the dataset.

The study by Golub *et al.* in [GOL99] uses Self-Organizing Maps (SOMs) to group Leukemia dataset. In this approach, the user specifies the number of clusters to be identified. The 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 [GOL99] got two clusters acute myeloid leukemia (AML) and acute lymphoblastic leukaemia (ALL), as well as the distinction between B-cell and T-cell ALL, which means that the optimal number of clusters is 2 or 3 (with subclasses).

The proposed genetic algorithm-based approach has been 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 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, 10] was picked for finding the optimal number of clusters.

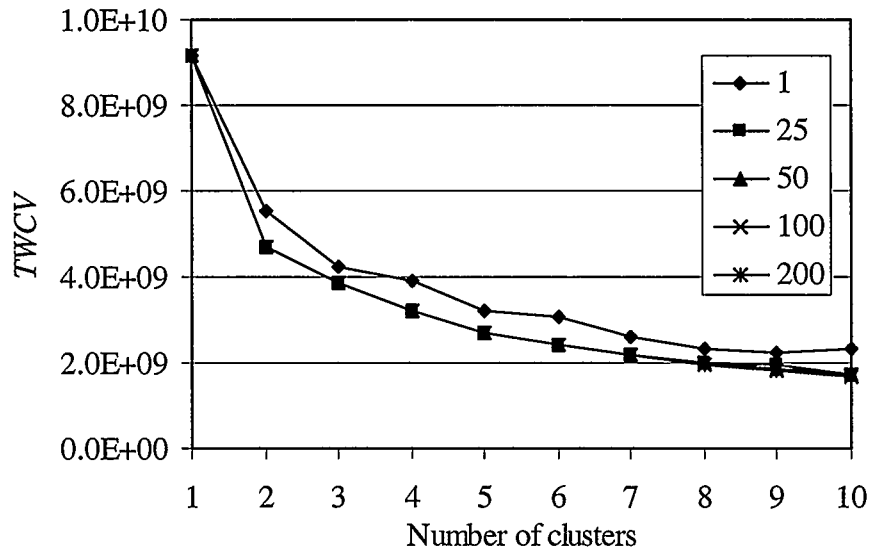


Figure 4.14. Pareto-fronts for Leukaemia dataset

Changes in the Pareto-optimal front are displayed in Figure 4.14 and Table 4.5 for different generations. It demonstrates how the system converges to an optimal Pareto-optimal front.

Table 4.5 Leukaemia Dataset *TWCV* for  $k = 9$

Iteration	<i>TWCV</i>
1	2.25E+09
25	1.94E+09
50	1.88E+09
100	1.84E+09
200	1.81E+09

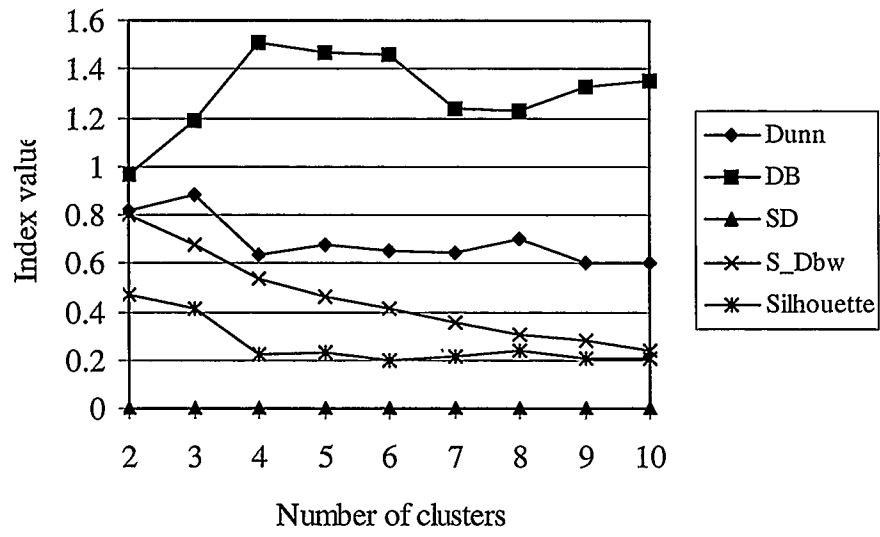


Figure 4.15 Leukemia dataset cluster validity results using Dunn, DB, SD, S\_Dbw and Silhouette indices



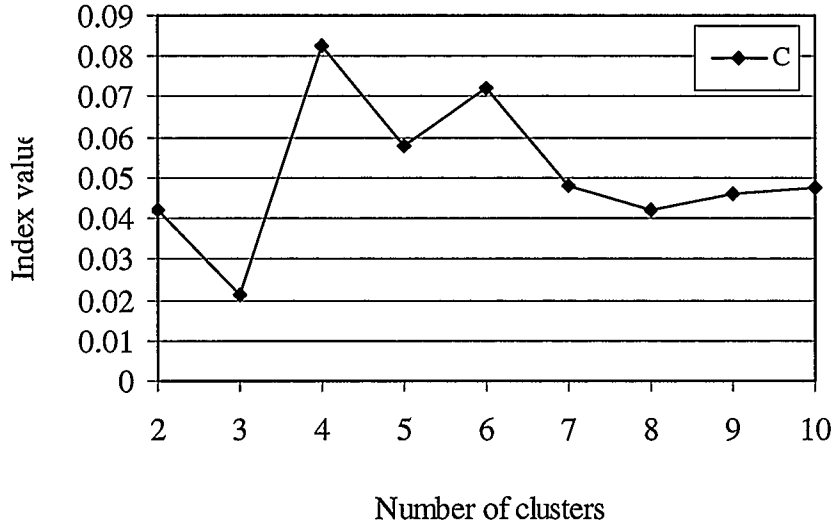


Figure 4.16 Leukemia dataset cluster validity results using C index

The Leukaemia Dataset clustering results in this thesis shown in Figure 4.15 and Figure 4.16 indicate the same conclusions as in [GOL99] by Golub *et al.*. 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 cluster number and 3 is the second. It can be seen from figure 4.15 that S\_Dbw is an exception. SD index gives good values but S\_Dbw does not. This indicates that the inter-cluster density for number of clusters taken 2 and 3 is not high for the 38 samples. Experimental results in this thesis also indicate S\_Dbw indices is not suitable to test small datasets with fewer than 40 instances.

### 4.3 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 five

datasets. In this section, a general evaluation is given, and the MOKGA system is compared with other methods on the basis of the results obtained from the same datasets.

#### 4.3.1 General evaluation

The Ruspini dataset clustering result shows that four is the optimal cluster number in all the cluster validity analysis indexes. This is consistent with earlier result [RUS70]. The Iris dataset gives similar result with the solutions of having the number of clusters two as the best solution and 3 the second best. According to [VRI99], Fig2data has 10 clusters. The proposed approach gave the same result using C index clustering validity method. Cancer data has 15 clusters according to the result in [US00]. MOKGA produces the same result using the DB index. The optimal number of clusters of Leukemia dataset is 2 or 3 (with subclasses). MOKGA reported the same results using Dunn, DB, SD, and Silhouette indices.

#### 4.3.2 Comparisons with other methods

##### **Multiobjective *K*-mean Genetic Algorithm (MOKGA) Vs. Fast Genetic *K*-mean Algorithm (FGKA):**

Since MOKGA has been developed on the basis of Fast Genetic *K*-mean Algorithm (FGKA) [YLU04] 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 Total Within-Cluster Variation (TWCV) for the fitness value evaluation.

According to the results, MOKGA and FGKA got similar TWCV values, MOKGA obviously need more generations to get the stable state, this might be because MOKGA 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 of gene expression data. These two issues are real concerns for FGKA, GKA and most of the other clustering algorithms.

#### **Multiobjective K-mean Genetic Algorithm (MOKGA) Vs. Neighborhood Analysis:**

The study in [GOL99] uses Self Organizing Maps (SOM) to group Leukemia dataset. Their method gets 2 classes, and for each of them, they get 2 subclasses. Exactly the same results are obtained in this thesis except for S\_Dbw. Experiments in this thesis indicate that the index is not suitable to test small datasets, like when number of instances is less than 40. In the experiment conducted for the study described in [GOL99], they used SOM method with user defined number of clusters, whereas the method proposed in this thesis does not need such value predefined.

#### **Multiobjective K-mean Genetic Algorithm (MOKGA) Vs. Average-linkage algorithm:**

The study described in [US00] uses an average-linkage algorithm and a metric based on the cancer dataset. A correlation parameter was applied to control the clustering results.

This parameter might be difficult to decide if it is an unsupervised clustering task. The number of clusters 15 was obtained in this thesis. It ranks the first for overall performance in DB index. This is consistent with the result reported in [US00].

#### **Multiobjective K-mean Genetic Algorithm (MOKGA) Vs. Visual Rendering:**

Keke Chen applied Visual rendering clustering algorithm on Iris dataset in 2003. 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, the researcher 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 3 clusters. Results also clearly show that separating them into 2 clusters is also reasonable. This can be verified from the map that the Visual rendering method delivered. In comparison to the Visual rendering method, MOKGA has the following advantages: it is more efficient, 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.

#### **Multiobjective K-mean Genetic Algorithm (MOKGA) Vs. Genetic Clustering Algorithm (GCA):**

In 1998, Rowena Marie Cole [RUS70] used a genetic algorithm (GCA) for clustering Ruspini dataset. We got the same clustering result they reported. Rowena's clustering system is similar to the proposed system in this thesis, they both have evolutionary based clustering algorithm and clustering validity methods, but the GCA cannot find Pareto optimal front in one run, and the process is relatively complex.

## Chapter Five

### Discussions and Conclusions

This thesis investigates the clustering approaches in general and investigates their applicability for clustering gene expression datasets, which including hierarchical clustering [JAH75], partitional clustering [TKO97] and recently emerging clustering techniques such as graph-based [BSY99] and model-based [KYY01] [YB02] approaches. Some traditional clustering algorithms which have been used for clustering gene expression datasets have also been discussed, including *K*-Means, Self Organizing Maps (SOM), heriatical clustering method, model-based approaches like Bayesain method, and the mixed model-based clustering algorithms.

A multi-objective genetic algorithm called MOKGA is proposed in this thesis to handle the expression data clustering problems. It is developed on the basis of the Niche Pareto optimal and fast *K*-means genetic algorithm. By using MOKGA, it is aimed at finding the Pareto-optimal front is sought to help the user to achieve 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 proposed clustering approach are demonstrated by conducting experiments using five datasets: figure2data, cancer (NCI60) and Leukaemia, Iris and the Ruspini.

In MOFGA, both crossover and mutation operators are used for the evolutionary process, in addition to the  $K$ -means operator are used to make the evolutionary process faster. For the selection, Niche 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 solutions. In the MOKGA process, the distance (Euclidean distance) between the current generation's Pareto optimal front and the previous generation's is calculated and counted 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 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 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, Hierarchical clustering algorithms and GCA can not 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 Bayesian method and Mixed model-based clustering algorithms.

There are some possible areas of improvement for MOKGA. In this thesis, cluster validity techniques, including Silhouette, 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 [CB02], or using third criteria to test cluster validity.



## References

- [ALI00] A. Alizadeh *et al.* Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* 403, pp.503-511. February 03, 2000.
- [BA2003] N. Bolshakova, F. Azuaje, Improving expression data mining through cluster validation, Proceedings of the 4th Annual IEEE Conference on Information Technology Applications in Biomedicine. pp. 19-22. 2003.
- [BEG02] A. Ben-Hur, A. Elisseeff and I. Guyon. A stability based method for discovering structure in clustered data. To appear in the proceedings of PSB, pp.6-17. 2002.
- [BF01] Y. Barash & N. Friedman. Context-specific Bayesian clustering for gene expression data. Proc. of RECOMB, pp.12-21, 2001.
- [BG03] A. Ben-Hur and I. Guyon. Detecting Stable Clusters Using Principal Component Analysis. In Methods in Molecular Biology, M.J. Brownstein and A. Kohodursky (eds.) Humana press, pp. 159-182. 2003.
- [BRA00] A. Brazma, A. Robinson, G. Cameron, M. Ashburner. One-stop shop for microarray data. *Nature*. 403(6771): 699-700. 2000.
- [BSY99] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281. 1999.

- [CA02] C. Caldas, and S.A.J. Aparicio. The molecular outlook. *Nature* 415(Jan. 31):484-485. 2002.
- [CB02] K. Curtis & M. Brand, Control analysis of DNA microarray expression data, *Mol Biol Rep.* 29(1-2):67-71. 2002.
- [CFP93] CM Fonseca and PJ Fleming, Genetic Algorithms for Multi-objective Optimization: Formulation Discussion and Generalization, *Proceedings of the 5th International Conference on Genetic Algorithms*, pp.416-423, June 01, 1993.
- [CL03] K. Chen, L. Liu: Validating and Refining Clusters via Visual Rendering. *Gene Expression Data of the Genomic Resources*, University of Stanford. *ICDM*: 501-504. 2003.
- [CPM02] B.A. Cohen, Y. Pilpel, R.D. Mitra, and G.M. Church, Discrimination Between Paralogs Using Microarray Analysis: Application to the Yap1p and Yap2p Transcriptional Networks. *Mol. Biol. Cell*, 13:1608-1614. 2002.
- [CW03] S. Cho, and H. Won, Machine Learning in DNA Microarray Analysis for Cancer Classification. In *Proc. First Asia-Pacific Bioinformatics Conference (APBC2003)*, Adelaide, Australia. *Conferences in Research and Practice in Information Technology*, 19. Chen, Y.-P. P., Ed. ACS, pp. 189-198. 2003.
- [DE02] E. Domany, Cluster Analysis of Gene Expression Data, *physics* 110, 1117. 2002.

- [DSY99] B. Dor, R. Shamir, and Z. Yakhini, Clustering gene expression patterns, *Journal of Computational Biology*, 6(3/4):281--297,1999.
- [DUG99] D. Duggan, et al. Expression profiling using cDNA microarrays. *Nat. Genet.* 21:10-14. 1999.
- [EC99] R. Ekins and F. Chu. Microarrays: their origins and applications. *TIBTECH*, 17:217-218, June 1999. CBC NOTE: in CR library.
- [EZI99] 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, pp. 19-39,1999.
- [GD03] B. Gary Fogel et al. *Evolutionary Computation in Bioinformatics*, pp. 219-225. 2003.
- [GMC02] R. A. Gutierrez, R. M. Ewing, J. M. Cherry, and P. J. Green. Identification of unstable transcripts in Arabidopsis by cDNA microarray analysis: Rapid decay is associated with a group of touch- and specific clock controlled genes. *PNAS*, August 20, 99(17): 11513 - 11518. 2002.
- [GOL99] T. R., Golub, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286: 531-7. 1999.
- [GRA98] N.S. Gray, et al. Exploiting Chemical Libraries, Structure, and Genomics in the Search for Kinase Inhibitors. *Science*, 281: 233-238. 1998.

- [GST99] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, and E.S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring: Science, 286:531--537, 1999.
- [HBV02] M. Halkidi, Y. Batistakis, M. Vazirgiannis: Clustering Validity Checking Methods: Part II. SIGMOD Record 31(3): 19-27. 2002.
- [HN93] J. Horn, N. Nafpliotis, Multi-objective optimization using the niched Pareto genetic algorithm. IlliGAl Rep. 93005. University of Illinois at Urbana-Champaign, Champaign, IL. 1993.
- [HNG94] 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.
- [HUG00] T.R. Hughes, et al. Functional discovery via a compendium of expression profiles. Cell 102, pp. 109-126. 2000.
- [HWJ02] H. Wang, W. Wang, J. Yang, S. Yu, Clustering by Pattern Similarity in Large Data Sets, Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 394-405. 2002.

- [JAH75] J.A. Hartigan, Clustering Algorithms: New York: John Wiley and Sons, pp. 353. 1975.
- [JGR03] J. Grabmeier, et al, Techniques of Cluster Algorithms in Data Mining, Kluwer Academic Publishers, Data Mining and Knowledge Discovery, Vol.6, pp. 303-360, 2003.
- [JMA65] J. MacQueen, Some methods for classification and analysis of multivariate observations, Proceedings of Berkeley Symp Math Stat Probability (Edited by: University of California Press). Cam LML, Neyman J, pp. 281-297. 1965.
- [JTZ03] D. Jiang, C. Tang, and A. Zhang. Cluster Analysis for Gene Expression Data: A Survey. IEEE Transactions on Knowledge and Data Engineering, pp. 1-5. 2003.
- [KC01] M. Kathleen Kerr and G. Churchill.. Bootstrapping cluster analysis: assessing the reliability of conclusions from microarray experiments. PNAS, 98:8961-8965, 2001.
- [KJA99] K. Jain, et al, Data Clustering: A Review, ACM Surveys, Vol.31, No.3, pp. 264 - 323. 1999.
- [KM99] K. Krishna and M. Murty, Genetic K-means algorithm, IEEE Transactions on Systems, Man, and Cybernetics - PartB: Cybernetics, 29(3):433---439, 1999.

- [KYY01] K.Y. Yeung, et al, Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17, pp.977-987, 2001.
- [LD01] E. Levine, E. Domany. Resampling Method for Unsupervised Estimation of Cluster Validity. *Neural Computation* 13: 2573–2593, 2001.
- [LG03] W. Li, I. Grosse Gene selection criterion for discriminant microarray data analysis based on extreme value distributions. *RECOMB*; pp.217-223. 2003.
- [LH02] V. LJ, D. H et al. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*; 415:530-536. 2002.
- [LOAB04] Y. Liu, T. Özyer, R. Alhajj and K. Barker, “Multi-objective Genetic Algorithm based Clustering Approach and Its Application to Gene Expression Data,” *Proceedings of Biennial International Conference on Advances in Information Systems*, Springer-Verlag, Izmir, Turkey, Oct. 2004.
- [LS02] L. Shi. DNA Microarray (Genome Chip) --- Monitoring the Genome on a Chip. <http://www.gene-chips.com/>. (c) 1998-2002.
- [MA95] B.J.T. Morgan, and A.P.G. Ray, Non-uniqueness and Inversions in Cluster Analysis. *Applied Statistics*, **44**(1): pp. 117-134. 1995.
- [MD01] *Microarray Data Analysis: Direct Gene Sample Correlations*, Gene Network Science, Inc. (c). 2001.

- [MJL01] P. McConnell, K. Johnson, D. J. Lockhart. An introduction to DNA microarrays. CAMDA (Conference) (2nd : October 15-16, Duke University Medical Center). 2001.
- [MM03] J. C. Mar, G. J. McLachlan: Model-Based Clustering in Gene Expression Microarrays: An Application to Breast Cancer Data. APBC: 139-144. 2003.
- [MRT03] U. Möller, D. Radke, F. Thies. Testing the significance of clusters found in gene expression data. European Conference on Computational Biology ECCB2003. Paris, 26-30, 9. 2003.
- [ND99] M. Neef, D. Thierens, & H. Arciszewski, A Case Study of a Multi-objective Elitist Recombinative Genetic Algorithm with Coevolutionary Sharing. In Angeline, P (Ed.), Proceedings of the International Congress on Evolutionary Computation, pp. 796-803. Priscataway: IEEE Press. 1999.
- [OLAB04] T. Özyer, Y. Liu, R. Alhajj and K. Barker, "Validity Analysis of Clustering Obtained Using Multi-Objective Genetic Algorithm," Proceedings of the International Conference on Intelligent Systems Design and Applications, Springer-Verlag, Hungary, Aug. 2004.
- [PS87] 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.

- [PT99] P. Tamayo, et al, Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, Proceedings of. Nat'l Acad Sci USA, 96, pp.2907-2912, 1999.
- [RBC03] F. Rioult, J. Boulicaut, B. Crémilleux, and J. Besson. Using transposition for pattern discovery from microarray data. In Proceedings of the ACM SIGMOD Workshop DMKD'03, San Diego, USA, pp. 73 - 79, June 2003.
- [RLBB02] V. Roth, T. Lange, M. Braun, M. Buhmann. A Resampling Approach to Cluster Validation. Computational Statistics - COMPSTAT, Physica Verlag. pp.123-128. 2002.
- [ROW98] R. M. Cole. Clustering with genetic algorithms. [http://www. cs. uwa. edu. au/pub/robvis/theses/RowenaCole](http://www.cs.uwa.edu.au/pub/robvis/theses/RowenaCole). 1998.
- [RSI02] M. Ramoni, P. Sebastiani and I.S. Kohane. Cluster Analysis of Gene Expression Dynamics. Proc Nat Acad Sci USA. 99(14):9121-6. 2002.
- [RUS70] E. H. Ruspini, Numerical methods for fuzzy clustering. Inform. Sci., 2, 319–350. 1970.
- [SCD03] W. Shannon, R. Culverhouse J. Duncan. Analyzing microarray data using cluster analysis. Pharmacogenomics, 4(1):41-52. 2003.
- [SCH85] J.D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: J.J. Grefenstette (ed.), Genetic Algorithms and Their



Applications: Proceedings of the Third International Conference on Genetic Algorithms, Lawrence Erlbaum, Hillsdale, NJ, 93-100. 1985.

- [SD94] N. Srinivas and K. Deb, Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computer*. 2,3, pp. 221–248. 1994.
- [SG03] M. Smolkin, and D. Ghosh, Cluster stability scores for microarray data in cancer studies. Technical Report, pp.4-36. 2003.
- [SJ02] Shah, J. Harendra: A Review of DNA Microarray Data Analysis. Final Projects Submitted for Credit in Computational Molecular Biology, Biochemistry 218/ Medical Information Sciences 231, 2002.
- [SK02] E. Segal, D. Koller: Probabilistic hierarchical clustering for biological data. In Proc. 6th Inter. Conf. on Research in Computational Molecular Biology (RECOMB), Washington, DC, pp. 273--280. April 2002
- [SK97] S. Kaski, Data exploration using Self-Organizing maps. *Acta Polytechnica Scandinavica, Mathematics, Computing and Management in Engineering Series No. 82*, pp 57. March 1997.
- [SMW03] B. Stein, S. Meyer and F. Wissbrock. On Cluster Validity and the Information Need of Users. In the Proceedings of the 3rd IASTED International Conference on Artificial Intelligence and Applications (AIA'03), Benalmadena, Spain, September 08-10, 2003.

- [SRW00] U. Scherf, et al. A Gene Expression Database for the Molecular Pharmacology of Cancer: Nat Genet 24, pp.236-244. 2000.
- [SS01] R. Shamir and R. Sharan, Algorithmic approaches to clustering gene expression data: Current Topics in Computational Biology, MIT Press, pp. 269-299, 2001.
- [TKO97] T. Kohonen, Self-organizing Maps: Berlin/Heidelberg: Springer-Verlag, 1997.
- [TRG99] T.R. Golub, et al, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, 286, pp.531-537, 1999.
- [TWH01] R. Tibshirani, G. Walther, T. Hastie. Estimating the number of clusters in a data set via the gap statistic. JRSS-B, 63, pp.411-423. 2001.
- [UF02] R. Ulrich, and S. Friend, Toxicogenomics and drug discovery: will new technologies help us produce better drugs? Nat. Rev. Drug Discov. 1: 84-88, 2002.
- [US00] U. Scherf, et al, A Gene Expression Database for the Molecular Pharmacology of Cancer, Nat Genet 24, pp.236-44, 2000.
- [VP1896] V. Pareto. Cours d'economic politique. Dronz, Geneva Switzerland, 1896.
- [VRI99] V.R. Iyer, et al, The transcriptional program in the response of human fibroblasts to serum, Science 283(5398), pp.83-7, 1999.

- [WK00] P.J. Waddell, H. Kishino: Cluster Inference Methods and Graphical Models Evaluated on NCI60 Microarray Gene Expression Data. *Genome Informatics* 11, pp.129-140. 2000.
- [WRM00] D. Wang, H. Resson, M. Musavi, C. Domnisoru; Double Self-Organizing Maps to Cluster Gene Expression Data. *ESANN'2000 proceedings*. ISBN 2-930307-02-1, pp. 45-50. 2000.
- [WY03] S. Wu, H. Yan: Microarray Image Processing Based on Clustering and Morphological Analysis. *APBC*: 111-118. 2003.
- [YB02] Y. Barash, Context-Specific Bayesian Clustering for Gene Expression Data. *Journal of Computational Biology*, **9**:169-191, 2002
- [YLU04] Y. Lu, et al, FGKA: A Fast Genetic K-means Clustering Algorithm, *Proceedings of ACM Symposium on Applied Computing*, Nicosia, Cyprus, pp.162-163, 2004.

## Appendix A

### Cluster validity results

#### 1. Cluster validity results for Leukaemia dataset

<b>Number of clusters</b>	<b>Dunn</b>	<b>DB</b>	<b>SD</b>	<b>S_Dbw</b>	<b>Silhouette</b>	<b>C</b>
<b>2</b>	0.816302	0.963104	0.000294	0.801714	0.469944	0.042038
<b>3</b>	0.882528	1.18878	0.000289	0.677513	0.412883	0.021152
<b>4</b>	0.636145	1.51309	0.000336	0.53522	0.219935	0.082555
<b>5</b>	0.678682	1.46778	0.000331	0.462136	0.234052	0.057915
<b>6</b>	0.651348	1.45849	0.000347	0.411032	0.197411	0.072089
<b>7</b>	0.639514	1.234201	0.000302	0.351981	0.215911	0.047749
<b>8</b>	0.697644	1.232512	0.0003	0.302092	0.235597	0.042049
<b>9</b>	0.601084	1.329422	0.000325	0.276681	0.203163	0.045872
<b>10</b>	0.599161	1.352388	0.000328	0.238616	0.207523	0.04753

## 2. Cluster validity results for Fig2data dataset

<b>Number of clusters</b>	<b>Dunn</b>	<b>DB</b>	<b>SD</b>	<b>S_Dbw</b>	<b>Silhouette</b>	<b>C</b>
<b>2</b>	0.757165	0.932779	-	-	0.776392	0.040804
<b>3</b>	0.904648	0.851353	1.772194	0.851742	0.738288	0.046575
<b>4</b>	0.445638	1.22227	1.733586	0.720494	0.538008	0.050832
<b>5</b>	0.326452	1.268101	2.258081	0.869284	0.509345	0.044687
<b>6</b>	0.297175	1.202612	2.062816	0.745646	0.486546	0.0424
<b>7</b>	0.20698	1.217804	2.756195	1.10103	0.509416	0.03035
<b>8</b>	0.231939	1.13895	2.155115	0.743688	0.49341	0.026079
<b>9</b>	0.219252	1.256533	2.011548	0.640726	0.485222	0.023815
<b>10</b>	0.185454	1.378938	2.250847	0.625845	0.455456	0.019997
<b>11</b>	0.185876	1.388135	2.04828	0.530905	0.458917	0.018416
<b>12</b>	0.231665	1.344745	1.766967	0.387019	0.462375	0.017079
<b>13</b>	0.274408	1.374182	1.506195	0.268499	0.465027	0.015018
<b>14</b>	0.268536	1.373209	1.524575	-	0.462974	0.013985
<b>15</b>	0.24915	1.351373	1.625935	0.23868	0.44799	0.013824
<b>16</b>	0.246076	1.501431	1.658829	-	0.45004	0.012644
<b>17</b>	0.233579	1.409388	1.852181	0.216034	0.439243	0.011287
<b>18</b>	0.203939	1.509113	2.202078	0.204957	0.406159	0.009704
<b>19</b>	0.2174	1.412827	2.133923	0.200268	0.412211	0.008122
<b>20</b>	0.167296	1.482442	2.669504	0.931215	0.394078	0.007489
<b>21</b>	0.266929	1.47751	1.7814	-	-	0.006547

### 3. Cluster validity results for cancer (NCI60) dataset

<b>Number of clusters</b>	<b>Dunn</b>	<b>SD</b>	<b>S_Dbw</b>	<b>Silhouette</b>	<b>DB</b>	<b>C</b>
<b>2</b>	0.806048	0.110659	0.938838	0.110364	2.60584	0.281144
<b>3</b>	0.829059	0.08715	0.817321	0.150427	2.1934	0.212971
<b>4</b>	0.773502	0.088299	0.803443	0.122332	2.27871	0.21199
<b>5</b>	0.703265	0.097601	0.791144	0.115684	2.8107	0.185538
<b>6</b>	0.708641	0.095058	0.822848	0.116664	2.523243	0.137107
<b>7</b>	0.681119	0.096103	0.811413	0.108295	2.514573	0.132263
<b>8</b>	0.671372	0.093632	0.765839	0.108287	2.436603	0.138629
<b>9</b>	0.671268	0.093795	0.718804	0.110278	2.40532	0.137473
<b>10</b>	0.69439	0.090476	0.704952	0.1191	2.235683	0.131666
<b>11</b>	0.68403	0.088327	0.642555	0.133249	2.187223	0.120912
<b>12</b>	0.65934	0.084985	0.605714	0.14278	2.16577	0.114811
<b>13</b>	0.721543	0.083808	0.592626	0.151204	2.20415	0.100341
<b>14</b>	0.761475	0.083444	0.604737	0.151678	2.128133	0.086632
<b>15</b>	0.767246	0.078352	0.475545	0.179657	2.068077	0.081889
<b>16</b>	0.793189	0.074822	0.455773	0.198995	2.068077	0.073042
<b>17</b>	0.793189	0.07047	0.404607	0.220043	2.068077	0.06975
<b>18</b>	0.793086	0.069992	0.383722	0.2331	2.03246	0.064866
<b>19</b>	0.778983	0.068771	0.361035	0.245958	2.048287	0.061703

#### 4. Cluster validity results for Ruspini dataset

<b>Number of clusters</b>	<b>Dunn</b>	<b>DB</b>	<b>SD</b>	<b>S_Dbw</b>	<b>Silhouette</b>	<b>C</b>
<b>2</b>	1.31725	0.724512	0.079156	0.351993	0.582726	0.027355
<b>3</b>	0.953401	0.532855	0.054054	0.15511	0.633405	0.051498
<b>4</b>	2.27981	0.374353	0.038952	0.048959	0.737657	0.001608
<b>5</b>	1.29515	0.500789	0.075495	0.036377	0.701924	0.003837
<b>6</b>	0.747283	0.817465	0.135803	0.119585	0.593999	0.020716
<b>7</b>	0.93904	1.11354	0.13389	0.100566	0.489046	0.019362
<b>8</b>	0.906991	0.871263	0.133282	0.106498	0.499191	0.02015
<b>9</b>	0.803495	0.87171	0.133163	0.088543	0.486977	0.021555

### 5. Cluster validity results for Iris dataset

<b>Number of clusters</b>	<b>Dunn</b>	<b>DB</b>	<b>SD</b>	<b>S_Dbw</b>	<b>Silhouette</b>	<b>C</b>
<b>2</b>	2.02591	0.404293	1.253414	0.788212	0.681046	0.02383
<b>3</b>	1.3881	0.661972	1.709599	0.737179	0.552819	0.03276
<b>4</b>	0.954282	0.787745	2.458602	0.913305	0.498051	0.02675
<b>5</b>	1.06648	0.839332	2.882346	0.789506	0.488781	0.02146
<b>6</b>	0.784807	1.023145	4.206412	0.879516	0.460521	0.02547
<b>7</b>	0.749252	1.098966	3.812975	0.721657	0.369526	0.02614
<b>8</b>	0.749864	1.108426	3.961877	0.669416	0.359924	0.02334
<b>9</b>	0.625991	1.121303	4.765977	0.638982	0.345266	0.02272
<b>10</b>	0.721925	1.14415	4.80393	0.584897	0.325041	0.02107