Multiobjective optimization of cluster measures in Microarray Cancer data using Genetic Algorithm Based Fuzzy Clustering

A thesis submitted in partial fulfillment
of the requirements for the degree of

Bachelor of Technology
in
Computer Science & Engineering

by

Shreeram Kushwaha
109CS0157

Under the guidance of

Prof. S.K. Rath

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India
Academic Year 2012-13

Certificate

This is to certify that the work in the thesis entitled *"Multiobjective optimization of cluster measures in Microarray Cancer data using Genetic Algorithm Based Fuzzy Clustering"* submitted by **Shreeram Kushwaha**, is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in the Department of Computer Science & Engineering, National Institute of Technology, Rourkela. Neither this thesis not any part of it has been submitted for any degree or academic award anywhere else.

Place:

Date:

Prof. Santanu K Rath
Professor
Dept. of Computer Science & Engineering
National Institute of Technology, Rourkela
Odisha - 769 008

Acknowledgement

No thesis is created entirely by an individual, many people have helped to create this thesis and each of their contribution has been valuable. My deepest gratitude goes to my thesis supervisor, Dr. Santanu K Rath, Professor, Department of Computer Science & Engineering, for his guidance, support, motivation and encouragement throughout the period this work was carried out. His readiness for consultation at all times, his educative comments, his concern and assistance even with practical things have been invaluable.

I would like to thank Miss Anita Ahirwar, for her valuable and important suggestions.

I must acknowledge the academic resources that we have acquired from NIT Rourkela. I would like to thank the administrative and technical staff members of the department who have been kind enough to advise and help in their respective roles.

I am thankful to all my friends. I sincerely thank everyone who has provided me with inspirational words, a welcome ear, new ideas, constructive criticism, and their invaluable time.

Last but not the least, I would like to dedicate this project to my family, for their love, patience and understanding.

Shreeram Kushwaha
Roll No.:109CS0157

Abstract

The field of biological and biomedical research has been changed rapidly with the invention of microarray technology, which facilitates simultaneously monitoring of large number of genes across different experimental conditions.

In this report a multi objective genetic algorithm technique called Non-Dominated Sorting Genetic Algorithm (NSGA) - II based approach has been proposed for fuzzy clustering of microarray cancer expression dataset that encodes the cluster modes and simultaneously optimizes the two factors called fuzzy compactness and fuzzy separation of the clusters. The multiobjective technique produces a set of non-dominated solutions. This approach identifies the solution i.e. the individual chromosome which gives the optimal value of the parameters.

Keywords: Fuzzy Clustering; Microarray expression data; Multiobjective Optimization

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Real world clustering problems are naturally posed as multiobjective, but evolutionary algorithms have been used in the past to optimize single objectives of clusters. As the name suggests, Multiobjective Optimization (MOO) is a process of optimizing systematically and simultaneously a collection of objective functions. For researchers and engineers, multiobjective optimization is very popular area for work. But it doesn't mean that there are no questions in this area, there are many open questions. There is no universally accepted definition of optimum, which makes it difficult to even compare the result of two methods, because the decision of best answer depends upon the decision maker. The potential to solve multiobjective optimization problem using evolutionary algorithm was hinted as early as the late 1960s by Rosenberg [1]. And the first actual implementation of Multiobjective Evolutionary Algorithm (MOEA) was produced until the mid-1980s by David Schaffer in his doctoral dissertation [2].

1.2 Clustering

Clustering an unsupervised classification technique, is the process of grouping or organizing a set of objects into distinct group based on some similarity or dissimilarity measure among the individual objects, such that the objects in the same group are more similar to each other than those in other groups.

It is mainly employed in Data Mining, and a common technique of statistical data analysis including the fields like Machine learning, Bioinformatics, Pattern Recognition, Image Analysis.

Considering Microarray expression data, clustering is an important microarray analysis tool, which is used to identify co-expressed genes i.e. genes with similar expression profiles.

Clustering can be mainly divided into two types Hard Clustering and Soft Clustering. They have been discussed in the next subsections.

1.2.1 Hard Clustering

Hard Clustering is based on classical set theory, and in this method of clustering the object either does or does not belong to a cluster [3]. In Hard clustering data is partitioned into specified number of mutually exclusive subsets.

1.2.2 Soft (Fuzzy) Clustering

In Soft Clustering [4], unlike hard clustering the object doesn't belong to a particular cluster rather an object belongs to more than one cluster simultaneously with different degree of membership and with every object there is an associated set of membership levels. The membership level indicates the strength of the association between that object and a particular cluster [5]. Objects on the boundaries between several classes are assigned a membership value between 0 and 1 indicating partial membership rather than they are not forced them to fully belong to a single cluster.

Using hard partitioning for algorithms based on analytic functional causes difficulties because hard partitioning is discrete in nature and also since this functional are not differentiable [6].

Unlike hard clustering, In fuzzy clustering, result is a $K * n$ membership matrix $U(X) = [u_{kj}], k = 1, ..., K$ and $j = 1, ..., n$, where u_{kj} denotes the probability of assigning x_j to cluster C_k. For probalistic non-degenerate clustering $0 < u_{kj} < 1$ and $\Sigma_{k=1}^{K} u_{kj} = 1, 1 \leq j \leq n$ [7].

1.3 Genetic Algorithm

Genetic Algorithm is a popular search heuristic which mimics the process of natural evolution and also it belong to the larger class of Evolution Algorithms. It is used to generate solutions to optimization problems using techniques inspired by natural evolution and selection to find the fittest individual in term of evolution. It is guided by the principle of Darwinian evolution, which considers a population which evolves in a particular environment, and only the fittest get a chance for reproduction and less fit solution got rejected due to environmental constraints. Genetic Algorithms find application in Bioinformatics, Computational Science, Economics, Chemistry and other fields.

1.4 Organization of the thesis

Chapter 1 Introduction

In this chapter general introduction about clustering and Genetic Algorithm is given.

Chapter 2 Literature Review

In chapter 2, What all work has been done is described.

Chapter 3 Multiobjective Optimization

Here In this chapter the concept of Multiobjective optimization and Pareto dominance has been described.

Chapter 4 Multiobjective Evolutionary Algorithms

In chapter 4, the detailed discussion about Evolutionary algorithms for multiobjective optimization has been done. And NSGA - II has been described in detail.

Chapter 5 Performance measure of NSGA - II

The performance measures of NSGA - II has been described.

Chapter 6 Application of NSGA - II in Microarray Gene Expression Data

In chapter 6, the application of NSGA - II on microarray expression data for optimizing cluster parameter is described.

Chapter 7 Simulation & Results

This chapter contains the simulation details and results of simulations.

Chapter 8 Conclusions

The overall conclusion of the thesis is presented in this chapter.

Chapter 2

Literature Review and Motivation

2.1 Literature Review

Genetic Algorithms have previously used for data clustering problems to develop efficient clustering techniques [8], [9] as earlier, generally Genetic algorithm has been used for optimizing a single objective, and that was not equally applicable for all class of datasets. And to solve problems, it might be some time necessary to optimize more than one objective simultaneously. Coming on to the problem of clustering, it is a real world problem, and clustering algorithms try to optimize validity measures like compactness, separation among the clusters or both. But to find out the relevance of different clustering criteria is unknown, so it is better to optimize the parameters separately rather than combining them into a single measure to get optimized.

There are instance in literature that applied multiobjective techniques for data clustering.

One of the recent approaches in this field is found in [10], where the objective functions representing separation and compactness of the clusters were optimized in a crisp clustering context and with a deterministic method.

In [11], a search based multiobjective criteria has been proposed, where the partitioning criteria chosen are the within-cluster similarity and between cluster dissimilarity and the technique used is based on cluster centers, as in [8].

In [12], [13] series of works on multiobjective clustering has been proposed, where chromosome encoding of length equal to number of data points. And the two objectives optimized were connectivity and compactness (overall deviation). And because of the length of chromosomes become equal to number of data points n to be clustered, the convergence become slower for large values of n [14] because of the chromosomes, and hence search space, in such cases become large.

As discussed in [14], when the length of chromosomes becomes equal to number of points n to be clustered, the convergence become slower for the large values of n. Because of the chromosomes and, hence search space, in such cases become large.

However in [13] a special mutation operator is used to reduce the effective search space by maintaining a list of L nearest neighbors for each data point, where L is user defined parameter. And this algorithm is intended for crisp clustering of continuous data.

2.2 Motivation

Clustering of microarray gene expression data is very important topic. Different clustering algorithms usually attempt to cluster the gene expression data but in this report multiobjective optimization of cluster validity measures such as compactness and separation among clusters in microarray cancer data has been proposed. The relative importance of different clustering criteria is unknown, so it is better to optimize compactness and separation separately rather than combining them into a single measure to be optimized.

The method proposed in this report uses a center(mode) based encoding strategy for fuzzy clustering of microarray cancer data.

And, computation of cluster modes is costlier than that of cluster means, the algorithm needs to find the membership matrices that takes a reasonable amount of time. However, as fuzzy clustering is better equipped with to handle overlapping clusters, the proposed technique can handle both overlapping and non-overlapping clusters. So, fuzzy K-mode has been used in this proposed work.

Chapter 3

Multiobjective Optimization

3.1 Overview

Many Real-world search and optimization problems in engineering are multiobjective in nature, because at the same time they normally have more than one objective that must be satisfied and those objectives may be possibly conflicting. And instead of finding single solution the term optimum needs to be redefined in context of multiobjective optimization, a set of good compromises or trade-offs needs to be produced and then the decision maker choose one out of those many solutions.

3.2 Definitions

3.2.1 Single Objective Optimization Problem (SOOP)

An optimization problem that involves optimization of single objective is known as Single Objective Optimization Problem.

In general a single objective function can be defined as minimizing or maximizing a function $f(x)$ subject to inequality constraints $g(x) \geq 0$, for all $i = 1, 2, ..., m$ and equality constraints $h(x) = 0$, for all $j = 1, 2, ..., p$, $x \in \Omega$. So, the solution minimizes or maximizes the function $f(x)$, where x is a n-dimensional decision vector variable $x = (x_1, ..., x_n)$ from some universe Ω.
The inequality and equality constraints must be fulfilled while optimizing (minimizing or maximizing) the objective function $f(x)$.

In SOOP, only a single optimal solution is obtained. And either the maximum or the minimum fitness value is selected as the optimal (best) solution depending upon the problem.

3.2.2 The Multiobjective Optimization Problems (MOP)

The process of optimizing a physical system, which involves a set of conflicting objectives subject to be optimized with certain constraints, is called MOP.

It can be defined as the problem of finding [15]: *"A vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term optimize means finding such a solution which would give the values of all the objective functions acceptable to the decision maker."*

The mathematical definition of a multiobjective problem (MOP) is important in providing a foundation of understanding between the interdisciplinary nature of deriving possible solution techniques (deterministic, stochastic); i.e. search algorithms [16].

The single objective formulation is extended to reflect the nature of multiobjective optimization problem where there is more than one objectives function which needs to be optimizing [16]. Thus there is set of solutions instead of a single solution i.e. a set of optimal solution and they are found using Pareto-optimality theory [428]. And a decision maker is required in multiobjective problems to make a choice of x_i^* values. The selection is necessarily to be tradeoff of one complete solution x over another in multiobjective space. Comparison between the set of solutions obtained is based on dominance and non-dominance.

In a precise manner, MOPs are those problems where the goal is to optimize k objective functions simultaneously. The set of k objective functions can be either all maximize or all minimize or combination of both. The objective functions can be linear or non-linear and continuous or discrete in nature. And also the objective function is a mapping from the vector of decision variables to output vectors. The decision variable can be continuous or discrete. General Definition of Multiobjective Optimization [17]

Finding the vector $\bar{x}^* = [x_1^*, x_2^*, ..., x_n^*]^T$ of the decision variables such that it will satisfy the m inequality constraints

$$g_i(\bar{x}) \geq 0, i = 1, 2, ..., m$$

and the p equality constraints

$$h_i(\bar{x}) = 0, i = 1, 2, ..., p$$

and optimizes the vector function

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), ..., f_i(\bar{x})]^T$$

The constraints define the feasible region F which contains all the allowable solutions [18]. Any solution which is outside this region is inadmissible since it violates one or more constraints [19]. Vector $x\bar{*}$ represents an optimal solution in F. In multiobjective optimization difficulty lies in the definition of optimality, since it is very rare that we will find a situation where a single vector $x\bar{*}$ will represent the optimum solution with respect to all the objective functions.

3.3 Pareto Terminology

The concept of Pareto optimality comes handy in the domain of multiobjective optimization problem.

Formal Definition of Pareto Optimality from the viewpoint of minimization problem:
A decision vector $\bar{x}^*$ is called Pareto optimal if and only if there is no $\bar{x}$ that dominates $\bar{x}^*$, i.e., there is no $\bar{x}$ such that

$$\forall i \in 1, 2, ..., k, f_i(\bar{x}) \leq f_i(\bar{x}^*)$$

and

$$\exists i \in 1, 2, ..., k, f_i(\bar{x}) < f_i(\bar{x}^*)$$

In general, Pareto optimum usually admits a set of solutions called the non-dominated solutions.

3.3.1 Dominance

A solution is said to dominate other if it is better in all objectives than the other solution. Mathematically, Solution vector $x = (x_1, x_2, ..., x_k)$ is said to dominate solution vector $y = (y_1, y_2, ..., y_k)$ if and only if x_i dominates y_i for all $i = 1, 2, ..., k$

3.3.2 Non-dominance

A solution is said to be non-dominated if it is better than the other solutions in atleast one objective. When Pareto points are plotted in objective space, the non-dominated solutions generates the pareto fronts.

3.4 Summary

This chapter summarizes the basic definitions and formal notation of general multi-objective optimization and concept of Pareto optimum that are adopted through the thesis.

Chapter 4

Multiobjective Evolutionary Algorithms

4.1 Overview

Evolutionary Algorithms (EA) are used to solve real-world multiobjective optimization problems due to their population based approach; a simple EA can be extended to maintain a diverse set of solutions. As evolutionary algorithms are population-based methods, it is easy to extend them to handle multiple objectives.

On the contrary, traditional search and optimization methods are difficult to extend to multiobjective case, since they generally deal with single solution. And due to the increasing interest to solve multiobjective problems, researchers have also developed new evolutionary algorithms based on real parameters. In this some of them are Non-Dominated Sorting Genetic Algorithm (NSGA), Pareto Archived Evolution Strategy (PAES) & Strength Pareto Evolutionary Algorithm (SPEA). If the problem is multiobjective then it gives rise to a set of optimal solutions known as Pareto Optimal Solution, instead of a single solution. By emphasizing one particular Pareto-Optimal Solution at a time, the classical optimization methods suggest to turn the multiobjective optimization problem to single objective optimization problem. When such methods are applied to find the solution, they are applied many times so that every time it results in hopefully a different solution. When emphasized on moving forward towards true pareto-optimal region, an EA can be used to find multiple Pareto-optimal solutions in one single run.

Two most desirable features of an Evolutionary Algorithm:

- Convergence to Pareto optimal front

  To achieve convergence to Pareto optimal front, most Multiobjective Evolutionary Algorithm (MOEA) use non-dominated sorting algorithms.
- Maintenance of Diversity (Representation of the entire Pareto optimal front) [20]

In this chapter, NSGA and NSGA - II has been discussed as how to apply it to multiobjective optimization.

4.2 Non-Dominated Sorting Genetic Algorithm (NSGA)

The Non-dominated sorting Genetic Algorithm is a popular non-domination based genetic algorithm for multiobjective optimization and is an instance of Evolutionary Algorithms. Actually NSGA is an extension of Genetic Algorithm for solving multiple objective function optimizations. It is related to other EAs of Multiobjective optimization like Strength Pareto Evolutionary Algorithm (SPEA), Pareto Archived Evolution Strategy (PAES).

NSGAs main objective is to improve the adaptive fitness of a population of candidate solutions to a Pareto front constrained by a set of objective functions. The algorithm uses an evolutionary process with surrogates for evolutionary operators including selection, genetic crossover, and genetic mutation. After that population is sorted based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions.

Classical NSGA and the updated & currently canonical form NSGA - II [16] are the two types of NSGA.Classical NSGA has been generally criticized for its computational complexity, lack of elitism and for choosing the optimal parameter value for sharing parameter σ_{share}.

4.3 NSGA - II

A modified and updated version of NSGA is called NSGA - II [21] was developed, it has better sorting, incorporates elitism and the sharing parameter need not to be chosen a priori. The elitism feature favors the elites of a population i.e. the non-dominated solution among the parent and child populations are directly propagated to the next generation. In this way a good solution found early will never be lost unless a better solution is discovered. The near-Pareto-Optimal string of the last generation provides different solutions to the clustering problem.

4.3.1 Fast Non-Dominated Sorting

Generally non-dominated sorting is one of the main time-consuming parts of multiobjective evolutionary algorithm (MOEA) [22]. So, design of fast non-dominated sorting algorithm is very necessary to improve the performance of a MOEA. And NSGA - II is a fast non-dominated Sorting Algorithm which has been used in this report.

In fast non-dominated sorting approach, the population is sorted based on non-domination. After initializing the population, it is sorted based on non-domination in each front. The first front being completely dominant in the current population, the individuals in the second front is only dominated by the individuals of first front and the front goes on. The individuals are assigned rank (fitness) values or based on front to which they belong. Individuals of first front are assigned rank 1 and individuals in second front are assigned a value of 2 and so on.

In addition to rank also a second parameter called crowding distance is calculated for every individual. Crowding distance measures how close an individual is to its neighbors. Large crowding distance will maintain a better diversity in the population.

The non-domination sorting is used in NSGA - II is fast because compare to other MOEAs, NSGA - II has been designed in such a way that the time complexity is small, hence the non-domination process is fast.

For population size of P and number of objective function O, fast non-dominated can defined as follow [21]

For each individual p, two entities are calculated

- Domination Count, n_p the number of individuals (solutions) which dominates the individual p, and
- S_p, a set of solutions which the individual p dominates.

All solutions in the first non-domination front will have $n_p = 0$. Then for every individual q in S_p, reduce the domination count by one and in doing so, if for any individual the domination count becomes zero then we put it into separate list Q, and the second front is identified. The process is continued until all fronts are identified.

The total complexity of the fast non-domination procedure is OP^2, whereas the complexity of normal non-domination sorting is OP^3.

4.3.2 Fitness Assignment Ranking Based on Non-Domination Sorting

Each individual of the population is assigned a rank (fitness) value based on the non-domination sorting procedure. After calculating the rank, for the individuals of same front crowding distance is also calculated.

4.3.3 Diversity Mechanism

The non-domination sorting algorithm converge the solution to the Pareto optimal front. But along with the convergence one more desirable feature of MOEA needs to be maintain, the diversity of the front i.e. a good spread of the solutions along the Pareto optimal front. The original NSGA uses a well-known sharing parameter which sets the desired extent of diversity. But this method makes the computation complex and also increased the dependence of the method on value of sharing parameter chosen. But In NSGA - II, the use of crowded comparison approach eliminated the above difficulties to some extent.

Density Estimation - Crowding Distance Assignment

Calculate the average distance of two points on either side of the point along each of the objective so as to get an estimate of the density of solutions surrounding a particular solution in the population. Crowding distance is assigned front wise and comparing the

crowding distance between two individuals in different front is meaningless. Crowding distance helps in obtaining uniform distribution.
The basic idea behind the crowding distance is finding the Euclidean distance between individual in a front based on their o objectives in the m dimensional hyper space. The individuals in the boundary are always selected since they have infinite distance assignment.

Crowded Operator based sorting

Crowded comparison operator ($\preceq_n$) is used to guide the process of selection at the various stages of the algorithm toward a uniformly spread-out Pareto optimal front. Assume that every individual i in the population has two attributes:

- Non-domination rank (i_{rank})
- Crowding Distance ($i_{distance}$)

Now, between two individuals i and j, the individual with lower rank will be selected(i.e. $i_{rank} < j_{rank}$) or if both individual belongs to the same front then their crowding distance is compared, and individual with greater crowding distance i.e. an individual located in a lesser crowded region is selected.

4.3.4 Elitism

The most characteristic part of NSGA - II is its elitism operation, where the non-dominated solutions among the parent and the child populations are propagated to the next generation.

4.4 Summary

The attractive feature of NSGA - II (MOEA) is their ability to find a wide range of non-dominated solutions which are close to the Pareto optimal solutions. Evolutionary algorithms process a population of solutions in every iteration, thereby making them ideal candidates for finding multiple trade-off solutions in one single simulation run.

Chapter 5

Performance Measure of NSGA - II

There exist many different MOEAs, so it is necessary that their performance needs to be quantified on a number of test problems. There are two goals in MOO, i.e. the performance evaluation of a multiobjective evolutionary algorithm is based on two metrics:

- Convergence Metric
- Diversity Metric

The performance parameters can be said that they are orthogonal to each other. The Convergence parameter search towards the Pareto-optimal region, while the diversity parameter requires a search along the Pareto-optimal front. They have described in the below sections.

5.1 Convergence Metric

The convergence metric γ measures the extent of convergence to a known set of Pareto-optimal solutions [21]. Convergence metric explicitly computes a measure of the closeness of set Q of N solutions from a known set of the true Pareto-optimal set P^*. Convergence Metric finds an average distance of Q, from P^*, as follow

$$\gamma = avg(\Sigma(mindistance))$$
$$\gamma = \frac{\Sigma_{i=1}^{N} d_i}{N}$$

where, d_i is the Euclidean distance (in the objective space) between the solution $i \in Q$ and the nearest member of P^*.

$$d_i = min_{k=1}^{|P|} sqrt(\Sigma_{m=1}^{M}(f_m^{(i)} - f_m^{*k})^2)$$

where, $f_m^{*(k)}$ is the m^{th} objective function value of the k^{th} member of P^*. When all the obtained solutions lies exactly on P^* chosen solutions, this metric takes a value of zero.

5.2 Diversity Metric

The diversity preservance mechanism avoids that the entire population converges to a single solution. Deb [21] the metric called the diversity metric, to measure spread of solutions obtained by an algorithm directly. The measure of diversity can be separated into two different measures of extent i.e. along the spread of extreme solutions and distribution i.e. the relative distance among the obtained solutions given by

$$\Delta = \frac{\Sigma_{m=1}^{M} d_m^e + \Sigma_{i=1}^{N} |d_i - \bar{d}|}{\Sigma_{m=1}^{M} d_m^e + N(d)}$$

where, d_i can be any distance measure between neighboring solutions and $\bar{d}$ is the mean value of these distance measure. The parameter d_m^e is the distance between the extreme solutions of P^* and Q corresponding to m^{th} objective function.

For the most widely and uniformly spread out set of non-dominated solutions, the numerator of Δ would be zero, making the metric value to zero. For any other distribution, the value of the metric would be greater than zero.

5.3 Summary

There are many other performance metrics but convergence metric and diversity metric are the two most important metrics to measure the performance of evolutionary algorithms like NSGA - II.

Chapter 6

Application of NSGA - II in Microarray Cancer Data

6.1 Overview

A microarray is an array of DNA molecules that permit many hybridization experiments to be performed in parallel. The progress in Microarray technology facilitates the monitoring of expression profile of a large number of genes across different experimental conditions simultaneously. Clustering of microarray gene expression data is used to identify the sets of co-expressed genes with similar expression profile.

A microarray gene expression data having g genes and h time points are typically organized in a $2D$ matrix $E = [e_{ij}]$ of size g x h. Each element e_{ij} gives the expression level of i^{th} gene at j^{th} time point. Microarray technology has many applications in the field of biological research, medical diagnostics, drug discovery and development, and toxicology [23]. In this report, a method has been proposed, which combines the feature of Multiobjective Genetic Algorithm based fuzzy clustering for optimization for fuzzy compactness and fuzzy separation of clusters of microarray cancer data.

6.2 Leukemia

Leukemia is a type of blood cancer characterized by an abnormal increase of immature white blood cells called blasts. Leukemia affects the bone marrow. The white blood cells help to fight infections and other diseases. Normally, the cell produce in an orderly way, but people that have leukemia the cell production gets out of control. The marrow produces too many immature white blood cells, which are differently shaped and cant carry out their usual duties. Leukemia is broad term covering range of diseases.

6.2.1 Dataset

The dataset that has been used in this work is probably the most famous gene expression cancer dataset (Golub et al.), containing information on gene-expression in samples from human acute myeloid (AML) and acute lymphoblastic leukemias (ALL). The original data set has 7129 genes and 72 samples but preprocessed data with 50 genes and 72 sample has been used.

6.3 Multiobjective Algorithm Non Dominated Sorting Genetic Algorithm - II

In this work NSGA - II is used as the multiobjective algorithm for optimization of cluster parameters. NSGA - II is well known multiobjective genetic algorithm which can maintain diversity on the Pareto front well. The chromosomes in this study are real coded.

The procedure of NSGA - II has been explained below:

Initialize the population by encoding K cluster modes in each chromosome by randomly selecting K objects from dataset. The process is repeated for every P chromosomes in the population, where P is the population size. Each chromosome is a sequence of attribute values representing the K cluster modes. Since K, cluster modes are encoded, and then the length of the chromosome will be K x p, where K is the number of clusters and p is the number of attribute in one sample. Then the values for the objective functions are calculated then the population is sorted based on non-domination in to front. The first front shows non-domination set in the current population and the second front being dominated by the individuals of the first front only and the front goes on. Each individual is assigned rank (fitness) values. In addition to rank, one more parameter called crowding distance is calculated for each individual. The selection method for selecting parents used here is Binary Tournament Selection based on rank and crowding distance. An individual is selected if its rank is lesser than the other and if both individuals have the same rank then their crowding distance is compared and the individual with larger crowding distance got selected. The selected individuals generate offsprings Q_t using simulated binary crossover and polynomial mutation operators. The population R_t generated by combining the current population Pt and the current offspring Q_t, is sorted again based on non-domination and only the best P individuals are selected, where P is the population size. This is repeated until the condition is met.

6.3.1 Optimization using NSGA - II

Genetic Algorithms have been previously used for clustering. And to solve real world problems, many times multiple objectives need to be optimized simultaneously. Clustering algorithms attempt to optimize the validity measures of a cluster such as compactness, separation among clusters.

The main contribution or objective of this work is to propose a multiobjective genetic algorithm (NSGA - II) based fuzzy clustering of Microarray Cancer data. NSGA - II will be used to optimize the two cluster validity measures compactness (fuzzy) and separation (fuzzy) of the clusters simultaneously. The chromosomes used in this study are real coded.

6.3.2 Problem Formulation

A constrained optimization problem can be formulated as follow:

$$\begin{gathered}\text{Minimize } f(\bar{x}) \\ \text{Subject to } g_j(\bar{x}) \geq 0, \quad j = 1, 2, ..., J \\ h_k(\bar{x}) = 0, \quad k = 1, 2, ..K\end{gathered}$$

Here, $f(\bar{x})$ is the objective function, $g_j(\bar{x})$ and $h_k(\bar{x})$ are the inequality and the equality constraints respectively.

Objective Functions in this Proposed work

$$\begin{gathered}\text{Minimize: Fuzzy Compactness } (\pi) = \Sigma_{i=1}^{K} \frac{\sigma_i}{n_i} \\ \text{Maximize: Fuzzy Separation } (Sep) = \Sigma_{i=1}^{K} \Sigma_{j=1, j \neq i}^{K} \mu_{ij} D(z_i, zj) \\ \text{Subject to} \quad 0 \leq u_{ik} \leq 1, \quad 0 \leq i \leq K, \quad 1 \leq k \leq n, \\ \Sigma_1^K u_{ik} = 1, \quad 1 \leq k \leq n, \\ \text{and} \\ 0 < \Sigma_1^n u_{ik} < n, 1 \leq i \leq K\end{gathered}$$

where,

σ_i is the variation of the i^{th} cluster.

n_i is the cardinality

μ_{ij} is the membership degree of each mode z_j encoded with respect to other encoded modes z_i in a chromosome

u_{ik} is the membership matrix

K is the number of clusters

n is the number of samples / objects in the dataset.

6.3.3 Computation of Objective function

A fuzzy clustering algorithm produces a membership matrix $U(X) = [u_{kj}], k = 1, ..., K$ and $j = 1, ..., n$, where u_{kj} denotes the probability of assigning object x_j to cluster C_k. The global compactness (π) [24] of the clusters and fuzzy separation Sep [24] have been considered the two objectives in this work, they need to optimize simultaneously. For computing the two measures, the modes encoded in the chromosome are extracted. Let these be denoted as $v_1, v_2, ..., v_K$. The membership matrix (U) is calculated as follow [25]:

$$U = [u_{ik}]$$

$$u_{ik} = \frac{1}{\Sigma_{j=1}^{K} \frac{D(z_j,x_k)}{D(z_j,x_k)}^{\frac{1}{m-1}}}, \quad \text{for } 1 \leq i \leq K, \quad 1 \leq k \leq n$$

where,

$D(z_i, x_k)$ and $D(z_j, x_k)$ are the dissimlarity measure between z_i & x_k and z_j & x_k, and m is the weighing coefficient.
[Note that while computing u_{ik} using equ, if $D(z_j, x_k)$ is equal to zero for some j, then u_{ik} is set to zero for all $i = 1, ..., K, i \neq j$, while u_{jk} is set equal to 1].

The variation σ_i and fuzzy cardinality n_i of the i^{th} cluster $i = 1, 2, ..., K$ are calculated using the equation [24]

$$\sigma_i = \Sigma_{k=1}^{n} u_{ik}^{m} D(z_i, x_k), \qquad 1 \leq i \leq K$$

and

$$n_i = \Sigma_{k=1}^{n} u_{ik}, \qquad 1 \leq i \leq K$$

So, global compactness π of the solution represented by the chromosome is then computed as [24]

$$\pi = \Sigma_{i=1}^{K} \frac{\sigma_i}{n_i}$$

To compute fuzzy separation we need the membership degree of each encoded mode with other modes encoded in that chromosome. Hence membership of each v_j to v_i, $j \neq i$ is computed as [24]

$$\mu_{ij} = \frac{1}{\Sigma_{l=1,l \neq j}^{K} \frac{D(z_j,z_i)}{D(z_j,z_l)}^{\frac{1}{m-1}}}, \quad i \neq j$$

Fuzzy Separation can be defined as [24],

$$Sep = \Sigma_{i=1}^{K} \Sigma_{j=1, j \neq i}^{K} D(z_i, z_j)$$

And in order to obtain compact cluster, compactness π should be minimized and to get well separated clusters, the measure fuzzy separation should be maximized [26].

6.3.4 Genetic Operators

Real coded GAs use Simulated Binary Crossover (SBX) [27], [28] operator for crossover and polynomial mutation [27], [29].

Simulated Binary Crossover. Simulated Binary Crossover simulates the binary crossover observed in nature and is given as below

$$c_{1,k} = \tfrac{1}{2}(1-\beta_k)p_{1,k} + (1+\beta_k)p_{2,k}]$$

$$c_{2,k} = \tfrac{1}{2}(1+\beta_k)p_{1,k} + (1-\beta_k)p_{2,k}]$$

where, $c_{i,k}$ is the i^{th} child with k^{th} component, $p_{i,k}$ is the selected parent and β_k (≤ 0) is a sample from a random number generated having the density

$$p(\beta) = \tfrac{1}{2}(\eta_c+1)\beta^{\eta_c}, \qquad \text{if } 0 \leq \beta \leq 1$$
$$p(\beta) = \tfrac{1}{2}(\eta_c+1)\tfrac{1}{\beta^{\eta_c}}, \qquad \text{if } \beta > 1$$

This distribution can be obtained from a uniformly sampled random number u between $(0,1)$. η_c is the distribution index for crossover. That is

$$\beta(u) = (2u)^{\frac{1}{\eta_c+1}}$$
$$\beta(u) = \tfrac{1}{(2(1-u))^{\eta_c+1}}$$

6.3.5 Polynomial Mutation

$$c_k = p_k + (p_k^u - p_k^l)\delta_k$$

where, c_k is the child and p_k is the parent with p_k^u being the upper bound on the parent component, p_k^l is the lower bound and δ_k is small variation which is calculated from a polynomial distribution by using

$$\delta_k = (2r_k)^{\frac{1}{\eta_m+1}} - 1, \text{ if } r_k < 0.5$$
$$\delta_k = 1 - (2(1-r_k))^{\frac{1}{\eta_m+1}}, \text{ if } r_k > 0.5$$

where, r_k is an uniformly sampled random number between $(0,1)$ and η_m is mutation distribution index.

6.4 Summary

This chapter summarise the procedure of NSGA - II for multiobjective optimization and give detailed description of Genetic operators used in NSGA - II.

Chapter 7

Simualtion and Results

7.1 Simulation

7.1.1 Hardware & Software Configuration

Hardware Configuration

- Operating System of Machine Windows 7 Professional
- RAM 3GB
- Processor Speed 2.40GhZ

For running the simulation the software used is

- MATLAB R2010b

7.1.2 Parameters Involved

Parameters	Value
Number of Generation	Variable
Population Size	100
Distribution Index for crossover η_c	20
Distribution index for mutation η_m	20
Number of Objectives	2
Pool Size	50
Tournament Size	2

The simulation has been run many time by fixing all of the parameters except one parameter i.e. number of generations. And results of simulations has been discussed in the next section.

7.2 Results

The graph plotted for the simulation shows the plot of the two objective function - compactness and separation on x-axis and y-axis repectively.
The graph plotted is of the pareto optimal solutions obtained.

7.2.1 Simulation 1

Number of Generations = 50

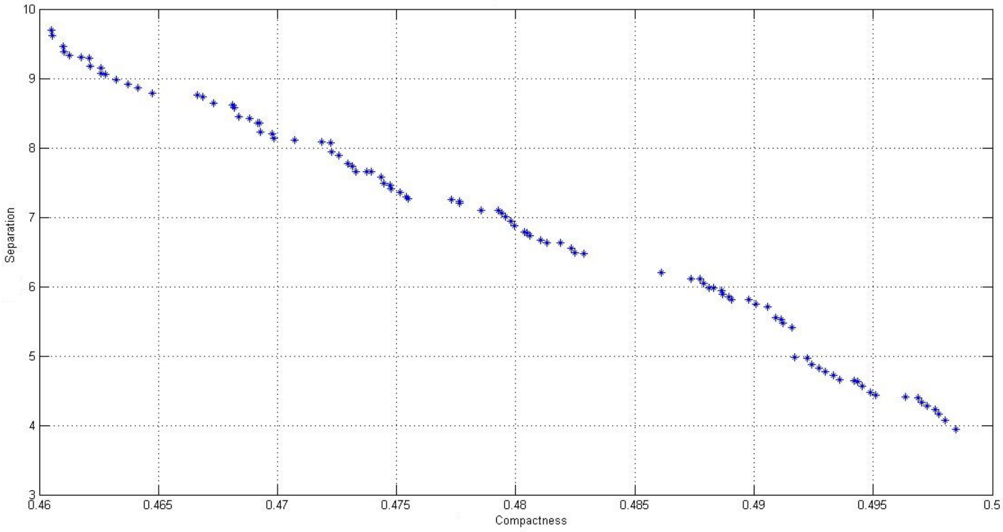

FIGURE 7.1: Pareto Optimal front solution, Generation = 50

Execution Time = 34.055 seconds

Optimal solution found at 1^{st} Chromosome with $K = 2$ modes with indices of the object 41, 14

Minimum Compactness = 0.4605 unit

Maximum Separation = 9.7004 unit

7.2.2 Simulation 2

Number of Generations = 100

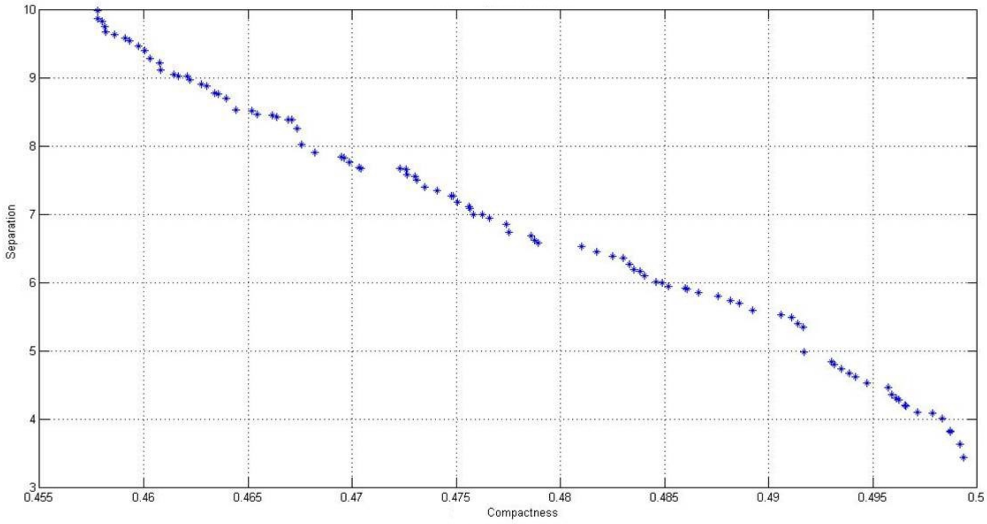

FIGURE 7.2: Pareto Optimal front solution, Generation = 100

Execution Time = 67.687 seconds

Optimal solution found at 2^{nd} Chromosome with $K = 2$ modes with indices of the object 57, 65

Minimum Compactness = 0.4578 unit

Maximum Separation = 9.9834 unit

7.2.3 Simulation 3

Number of Generations = 500

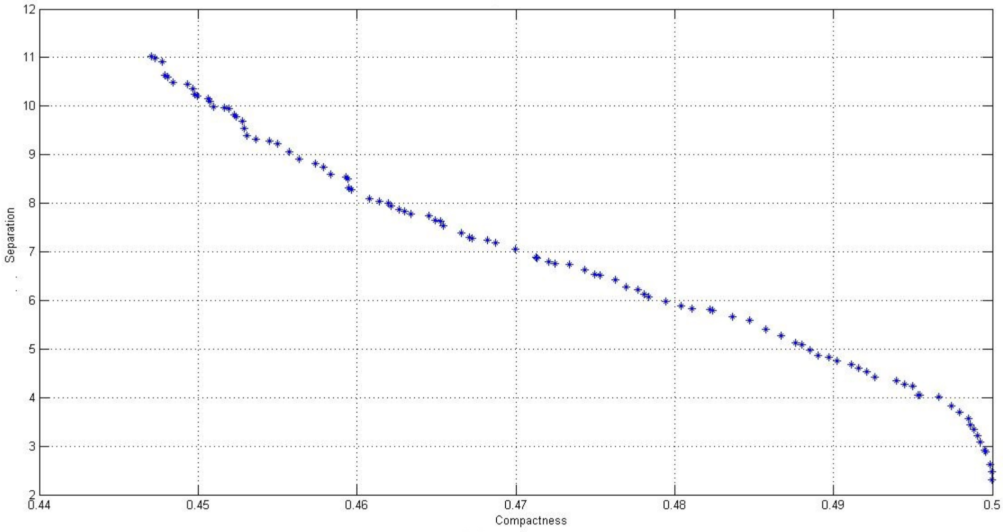

Figure 7.3: Pareto Optimal front solution, Generation = 500

Execution Time = 326.428 seconds

Optimal solution found at 1^{st} Chromosome with K = 2 modes with indices of the object 41, 14

Minimum Compactness = 0.4471 unit

Maximum Separation = 11.0226 unit

7.2.4 Simulation 4

Number of Generations = 1000

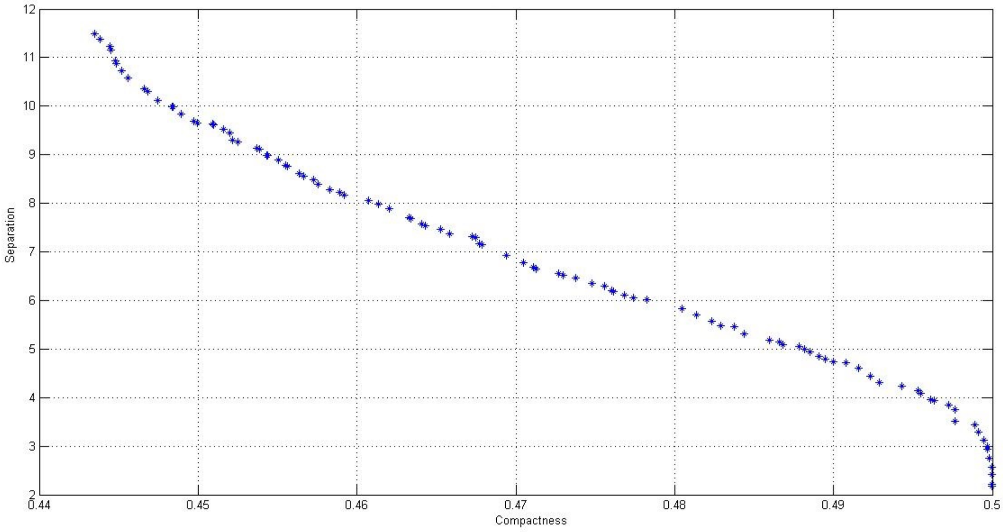

Figure 7.4: Pareto Optimal front solution, Generation = 1000

Execution Time = 667.170 seconds

Optimal solution found at 2^{nd} Chromosome with K = 2 modes with indices of the object 57, 65

Minimum Compactness = 0.4435 unit

Maximum Separation = 11.4837 unit

The property of the four plotted graph clearly shows that when compactness is minimized then separation between clusters maximizes.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The Algorithm NSGA - II for multiobjective optimization of cluster measures - compactness and separation for Leukemia cancer dataset, is working correct. From the simulations it is easily able to conclude that with the increase in the number of generations the solution is getting more and more optimized, it is optimizing both the objectives simulatneously. And the optimal values of compactness and separation are obtained at the same individual/solution which suggests the cluster validity measures are getting optimized.

8.2 Future Work

This proposed method can be applied on other type of microarray cancer data and other microarray gene expression data.

Bibliography

[1] Richard S. Rosenberg. Simulation of genetic populations with biochemical properties: Ii. selection of crossover probabilities. *Mathematical Biosciences*, 8(12): 1 – 37, 1970. ISSN 0025-5564. doi: 10.1016/0025-5564(70)90140-9. URL `http://www.sciencedirect.com/science/article/pii/0025556470901409`.

[2] James David Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms (artificial intelligence, optimization, adaptation, pattern recognition).* PhD thesis, Nashville, TN, USA, 1984. AAI8522492.

[3] J.V. Oliveira and W. Pedrycz. *Advances in fuzzy clustering and its applications.* Wiley, 2007. ISBN 9780470027608. URL `http://books.google.co.in/books?id=ZeFQAAAAMAAJ`.

[4] J. C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973. doi: 10.1080/01969727308546046. URL `http://www.tandfonline.com/doi/abs/10.1080/01969727308546046`.

[5] L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965. ISSN 0019-9958. doi: 10.1016/S0019-9958(65)90241-X. URL `http://www.sciencedirect.com/science/article/pii/S001999586590241X`.

[6] P.S. Szczepaniak, P.J.P.J.G. Lisboa, and J. Kacprzyk. *Fuzzy Systems in Medicine.* Dusseldorfer Kommunikations- Und Medienwissenschaftliche Stu. Physica-Verlag, 2000. ISBN 9783790812633. URL `http://books.google.co.in/books?id=Mvb13e8JZy0C`.

[7] James C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 1981. ISBN 0306406713.

[8] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm-based clustering technique. *Pattern Recognition*, 33(9):1455 – 1465, 2000. ISSN 0031-3203. doi: 10.1016/S0031-3203(99)00137-5. URL `http://www.sciencedirect.com/science/article/pii/S0031320399001375`.

[9] S. Bandyopadhyay and U. Maulik. Nonparametric genetic clustering: comparison of validity indices. *Trans. Sys. Man Cyber Part C*, 31(1):120–125, February 2001. ISSN 1094-6977. doi: 10.1109/5326.923275. URL `http://dx.doi.org/10.1109/5326.923275`.

[10] Michel Delattre and Pierre Hansen. Bicriterion cluster analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-2(4):277–291, 1980. ISSN 0162-8828. doi: 10.1109/TPAMI.1980.4767027.

[11] Rafael Caballero, Manuel Laguna, Rafael Mart, and Julin Molina. Multiobjective Clustering with Metaheuristic Optimization Technology.

[12] J. Handl and J. Knowles. Multiobjective clustering around medoids. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 632–639 Vol.1, 2005. doi: 10.1109/CEC.2005.1554742.

[13] J. Handl and J. Knowles. An evolutionary approach to multiobjective clustering. *Evolutionary Computation, IEEE Transactions on*, 11(1):56–76, 2007. ISSN 1089-778X. doi: 10.1109/TEVC.2006.877146.

[14] Sanghamitra Bandyopadhyay and Ujjwal Maulik. An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Inf. Sci. Appl.*, 146(1-4):221–237, October 2002. ISSN 0020-0255. doi: 10.1016/S0020-0255(02)00208-6. URL `http://dx.doi.org/10.1016/S0020-0255(02)00208-6`.

[15] C.A.C. Coello, G.B. Lamont, and D.A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Genetic and Evolutionary Computation. Springer, 2007. ISBN 9780387332543. URL `http://books.google.co.in/books?id=2murCij_wHcC`.

[16] J. Brownlee. *Clever Algorithms: Nature-inspired Programming Recipes.* Lulu Enterprises Incorporated, 2011. ISBN 9781446785065. URL `http://books.google.co.in/books?id=SESWXQphCUkC`.

[17] Carlos A. Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1:269–308, 1998.

[18] Luis Vicente Santana-Quintero and Carlos A. Coello Coello. An algorithm based on differential evolution for multi-objective problems, 2005.

[19] Ujjwal Maulik, Anirban Mukhopadhyay, and Sanghamitra Bandyopadhyay. Combining pareto-optimal clusters using supervised learning for identifying co-expressed

genes. *BMC Bioinformatics*, 10(1):1–16, 2009. doi: 10.1186/1471-2105-10-27. URL `http://dx.doi.org/10.1186/1471-2105-10-27`.

[20] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art, 2000.

[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6 (2):182–197, 2002. ISSN 1089-778X. doi: 10.1109/4235.996017.

[22] M. T. Jensen. Reducing the run-time complexity of multiobjective eas: The nsga-ii and other algorithms. *Trans. Evol. Comp*, 7(5):503–515, October 2003. ISSN 1089-778X. doi: 10.1109/TEVC.2003.817234. URL `http://dx.doi.org/10.1109/TEVC.2003.817234`.

[23] Shi Leming, Hu Weiming, Su Zhenqiang, Lu Xianping, and Tong Weida. Microarrays: Technologies and applications. In Dilip K. Arora and George G. Khachatourians, editors, *Fungal Genomics*, volume 3 of *Applied Mycology and Biotechnology*, pages 271 – 293. Elsevier, 2003. doi: 10.1016/S1874-5334(03)80016-3. URL `http://www.sciencedirect.com/science/article/pii/S1874533403800163`.

[24] George E. Tsekouras, Dimitris Papageorgiou, Sotiris Kotsiantis, Christos Kalloniatis, and Panagiotis Pintelas. Fuzzy clustering of categorical attributes and its use in analyzing cultural data. *International Journal of Computational Intelligence*, 1: 147–151, 2004.

[25] Zhexue Huang and M.K. Ng. A fuzzy k-modes algorithm for clustering categorical data. *Fuzzy Systems, IEEE Transactions on*, 7(4):446–452, 1999. ISSN 1063-6706. doi: 10.1109/91.784206.

[26] Anirban Mukhopadhyay, Ujjwal Maulik, and Sanghamitra Bandyopadhyay. Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. *Trans. Evol. Comp*, 13(5):991–1005, October 2009. ISSN 1089-778X. doi: 10.1109/TEVC.2009.2012163. URL `http://dx.doi.org/10.1109/TEVC.2009.2012163`.

[27] Kalyanmoy Deb and Ram Bhushan Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:1–34, 1994.

[28] H.-G. Beyer and K. Deb. On self-adaptive features in real-parameter evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 5(3):250–270, 2001. ISSN 1089-778X. doi: 10.1109/4235.930314.

[29] O. G. Kakde M. M. Raghuwanshi. Survey on multiobjective evolutionary and real coded genetic algorithms. pages 150–161. 8th Asia Pacific symposium on intelligent and evolutionary systems, 2004.