A VARIABLE-LENGTH MANY-OBJECTIVE OPTIMIZATION APPROACH IN IMAGE SEGMENTATION PROBLEMS

By

Xuhui Huang

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Electrical Engineering – Master of Science

2018

ABSTRACT

A VARIABLE-LENGTH MANY-OBJECTIVE OPTIMIZATION APPROACH IN IMAGE SEGMENTATION PROBLEMS

By

Xuhui Huang

Image segmentation is a technique of dividing an image space into a number of meaningful homogeneous regions. Various data clustering techniques have been adapted in solving segmentation problems. In particular, data clustering is often posed as multi-optimization problem so that characteristics of data could be caught by different objectives simultaneously. Traditional multi-optimization methods often require some prior knowledge or assumptions about data, performance is poor if these assumptions do not hold. Limitations with established multi-optimization methods are caused by their inadequacy in handling a large number of objectives. Nondominated sorting genetic algorithm III (NSGA-III) [1] is proposed to alleviate this issue. However, NSGA-III is inefficient in removing some bad solutions in high-dimensional searching space during evolution. In this article, we propose a variable string length many-objective genetic algorithm(VMOGA) whose framework has evolved from NSGA-III and its encoding strategy, genetic and evolutionary operator have been redesigned. Performance of VMOGA in image segmentation problems is further enhanced by an appropriate selection of objectives. In the end, we conduct unsupervised segmentation by proposed clustering technique on magnetic resonance image(MRI) of human brain. Comparisons with other evolutionary algorithms are presented and dominance of VMOGA has been demonstrated quantitatively. VMOGA is also performed on detection of delamination area caused by fatigue loading in Mode I glass fiber reinforced polymer (GFRP) samples. Results are compared with fast marching algorithm(FMA) and superiority of VMOGA suggests future potential application in fatigue detection.

ACKNOWLEDGEMENTS

I would first like to thank my advisor Yiming Deng at Michigan State University. The door to Prof. Deng office was always open whenever I ran into a trouble spot or had a question about my research or writing. I would also like to thank Prof. Udpa and Prof. Ulusoy in my thesis defense committee, without their passionate participation and comments, the validation of this thesis could not have been successfully conducted. I would also like to acknowledge Dr. Banerjee as the provider of the data to this thesis, and I am gratefully indebted to her for datailed explaination and guidance of the data.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Brief Introduction of Data Clustering

Data clustering has emerged as one of the fundamental technique for data analysis recently. One of the primary application of data clustering is image segmentation. The problem of image segmentation is usually considered as clustering image pixels in the intensity space [2] [1]. Geographic distribution sometimes is also incorporated to enhance homogeneity. Real-life applications of image segmentation include segmentation of remote sensing satellite images [3], magnetic resonance image(MRI) medical imagery of human body [4] and texture image [5]. Other applications such as social network analysis [6], time series data analysis [7], web mining [8] involve more sophisticated data. The purpose of clustering is to partition a given dataset into homogeneous groups. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with size n in a multi-dimensional space, the goal of clustering is to produce a partition matrix $U = \{u_{k,j}\}$, value of $u_{k,j}$ is the membership or belongingness of data point x_j to cluster C_k. Generally speaking, if $u_{k,j} = \max_{i \in [1,k]} \{u_{i,j}\}$ then $x_j \in C_k$. Crisp and fuzzy partition matrix both exist in the literature with their differences of selecting $u_{k,j}$. For crisp partition matrix, $u_{k,j} = 1$ or 0 while belongingness ranging from 0 to 1 in a fuzzy matrix [9]. It is worth to be noted that fuzzy clustering is better capable of handling overlapping and noisy clusters.

1.2 Multi-objective Optimization and Data Clustering

Performances of clustering solution are often evaluated by some validity indices. Therefore, clustering is posed as optimization problem by directly optimizing validity index [5] [6] [10]. Principle underlying this strategy is to consider partition matrix U equivalent to optimization searching space. Searching space is composed of all the possible clustering results which are denoted as Z. Notice that mapping between U and Z is unique and bilateral. On the other hand, there is a strong connection between performance of clustering results and validity index. This

is because validity index often represents properties of cluster such as homogeneity, separation, compactness, etc. In fact, when data is classified into reasonable or suitable groups, validity index is often optimized and vice versa. Further analysis on this transformation requires a thorough study on properties of cluster reflected by validity index, hence selection of validity index is important. There are some evident limitations with single-objective framework [11]. If only one validity index is optimized, characteristics of data are unlikely to be fully considered, thus adding complexity in discriminating two data groups with features alike. It is also extremely difficult to choose an objective with no prior knowledge about dataset. If wrong assumption is made at the beginning, clustering results would be corrupted [12]. In the end, it is impossible to implement tradeoff on objectives by single-objective optimization. At one time, in order to solve this issue, people adopt single-objective framework incorporating multiple objectives by ordering them according to importance or using weight vector to convert into single objective [13]. However, relative significance of various validity indices is unknown real-life application since data structure usually has not been specified. Moreover, it is extremely difficult to choose an objective with no prior knowledge about dataset. If wrong assumption is made at the beginning, clustering results would be corrupted. Therefore, optimizing multiple objectives simultaneously is demanded by situations.

1.3 Concept of Multi-objective Optimization

In order to achieve higher accuracy, our framework must optimize many objectives simultaneously. Multi-objective optimization can be formally stated as $\min_{x \in X}(f_1(x), f_2(x), \ldots, f_k(x))$, f_i denotes objective function. Maximum can be expressed as $\max_{x \in X} f_i(x) = \min_{x \in X}\{-f_i(x)\}$, where X is the feasible region. Solution to this problem is not unique. Moreover, good solution can be extracted by some standards. The most important notion here is Pareto optimality [09], if $x^* \in X$ is Pareto optimal, $\forall x \in X, s.t.$ (1) if $\forall i \quad f_i(x^*) \leq f_i(x), \exists i \in 1, 2, ...K, s.t \quad f_i(x^*) < f_i(x)$ (2) $\exists i \in 1, 2, \ldots, K$, $f_i(x^*) < f_i(x)$. Similarly, non-domination sorting rule can be written (in minimization style): if $\forall i \in 1, 2, ..., K, f_i(x) \leq f_i(y)$, $\exists i \in 1, 2, ...K, f_i(x) < f_i(y)$ then x dominate y

1.4 Genetic Algorithms(GAs)

One of the most efficient approach in searching solutions of Pareto optimality in multi-optimization problem is Genetic Algorithms(GAs) [14], proposed by John Holland. It is proven that GAs are the most powerful searching technique in a large space. Inspired by evolutionary process theory created by Charles Darwin, GAs adopt some similar concepts such as parent, child, population selection, mutation, crossover. Initialization of GAs is to encode solutions(variables) into strings called chromosomes, by operating on chromosomes instead of variables. GAs conduct searching from multiple points instead of single point. Therefore, GAs are unlikely to get trapped in local optimum as other searching algorithms do. Each solution is then given a fitness value and evaluated based on fitness value that measures the goodness of the solution encoded in the chromosome. In this article, fitness value is equivalent to objective value. Thereafter, chromosomes are selected based on fitness value, those having better fitness are given more chance to be selected and reproduce other chromosome. Potential better solutions are reproduced from selected chromosomes by applying some operators such as mutation and crossover. The procedure that retain best chromosome and replace the worst during each generation is called elitism and better fitness than previous generation is provided. Model that build upon these notions are developing very fast in the last decade. Strength Pareto Evolutionary Algorithm (SPEA) [15] and [16], Pareto Envelope-Based Selection Algorithm (PESA) [17]and PESA-II [18], Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [19] are proposed and instantly become popular in application of multi-objective evolutionary algorithm for clustering problem.

1.4.1 Multi-objective Genetic Algorithm

In recent studies, multi-objective genetic algorithm (MOGA) based methods are frequently adopted in unsupervised image segmentation, medical and geographical detection problems. NSGA-II is one of the most widely used MOGA based methods since its proposal. However, procedures within NSGA-II framework that benefit its performance on two-objective optimization problems has

prevented it from achieving decent results on many-objectives problem (with number of objectives larger than 3). NSGA-III was proposed to alleviate the issue, as an evolutionary many-objective algorithm, it becomes popular instantly. As a reference-point-based searching algorithm, solution in each iteration is encouraged to be close to vacant reference lines, thus adding coverage among searching space.

1.4.2 General Framework of MOGA

Existing MOGA techniques mainly differ in five aspects [20]. First of all, data clustering is posed as optimization problems by optimizing some validity indices (objective functions). Therefore, selection of objective function must be done in the first place. Then, cluster solution will be represented in the form of chromosome. Encoding operators are different in terms of representation strategy. Among them, label representation and coordinate representation from Label-Based Encoding and Prototype-Based Encoding respectively are most commonly applied. Cluster solution is initialized as parent solution population in this period. In the next stage, genetic operator produces child solution population. Genetic operator is highly related to encoding operator, so they are often discussed together. The key aspect of MOGA is the evolutionary operator which makes selection on parent-child combined population and generates solution population for the next generation. In some articles, genetic operator and evolutionary operator are bounded together as evolutionary operator. Here, we want to stress the importance of selection strategy in evolution, thus, evolutionary operator is discussed separately. In the end, a single clustering solution must be obtained from the last generation. General framework of MOGA is illustated in fig[1.1]

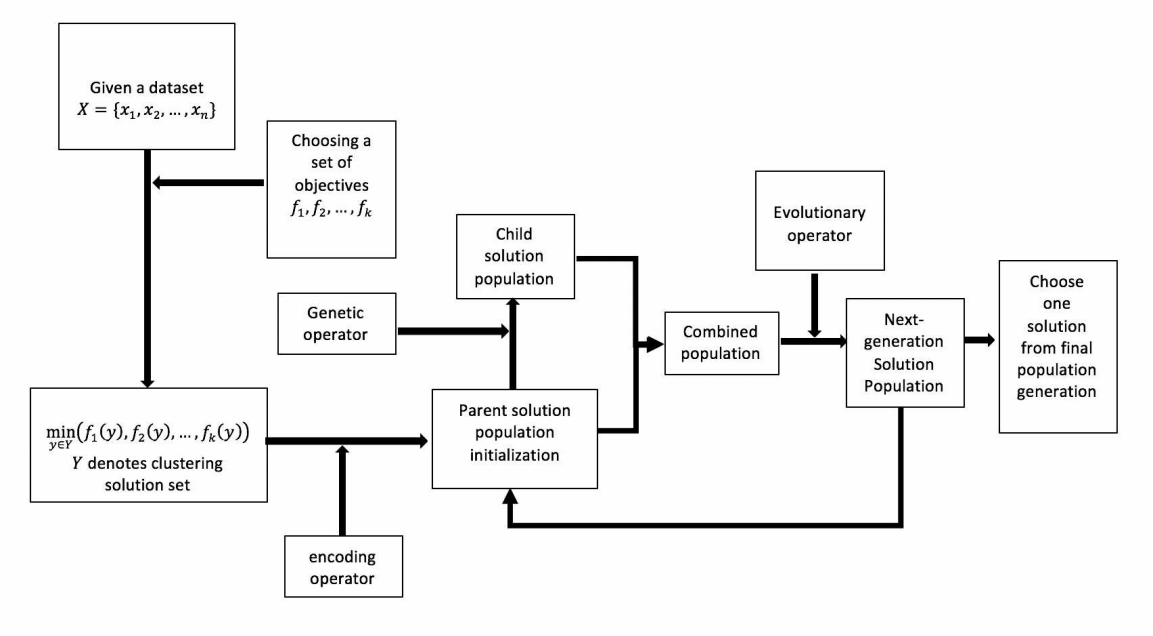

Figure 1.1 General framework of MOGA

1.5 Evolving the Number of Clusters

Conventional MOGA methods require predetermination of the number of cluster K. In fact, only an appropriate choice of K could yield acceptable clustering results. In this article, choice of the number of clusters can be tackled by evolving K automatically during iteration using variable string-length encoding in GAs. However, previous encoding strategies and proposed evolutionary operators [21] are inefficient to generate K since they ignore inherent distribution of binary mutation and crossover. In this paper, we examine child population distribution from different evolutionary operators. Performance of selecting K is also evaluated and superiority of proposed evolutionary strategy is demonstrated.

1.6 Structure of This Paper

This paper is organized as follows. In section 2, we introduce methods of constructing encoding and genetic operator. Inspired by some philosophy underlying conventional operator, we design a new strategy for VMOGA which is capable of handling variable number of. In section 3, evolutionary operator is introduced with respect to NSGA-II and III. Weakness with exiting strategy

is discussed and a selection scheme which remove "bad" solutions is proposed. In section 4, we discuss the choice of objective functions which are selected based on prior knowledge and supplement information provided by total correlation. In section 5, conventional objective functions are first introduced, then refined objective functions are constructed in order to better target image segmentation problem. In section 6, implementation details of proposed algorithm are presented. For MRI human brain image, evaluation of segmentation results which demonstrate superiority of proposed algorithm is accomplished by comparing adjusted rand index(ARI) and accuracy(ACC) with corresponding truth table. Results of experiment on detecting delamination area are also discussed.

CHAPTER 2

ENCODING STRATEGY AND GENETIC OPERATOR

2.1 General Encoding Strategy

Encoding strategy is adopted to represent a clustering solution. At first glance, clustering solutions can be represented by partition matrix. This representation scheme provides full coverage of entire solution space and it is convenient to define extra constraints during mutation or crossover. However, with increasing number of data points introduced in clustering problem, the complexity of direct encoding is surging exponentially, even for a mid-sized clustering problem, scale of its representation will be too large to handle. Consequently, reduction of searching space is vital for encoding and searching. Previous researches have chosen a more indirect approach: assigning each point to its closest cluster represented by centroid of the cluster and encoding coordinates of the centroids. Hence each representation has the length of $K \times d$, d is the dimension of data and K denotes the number of clusters. It is written as $[Z_1, Z_2, \ldots, Z_K]$, Z_i denotes center of the cluster which is vector of d by 1. In image segmentation, $d = 3$. Notice that K is a variable in VMOGA. Centroid-based encoding strategies produce considerably smaller scale of representation and reduce its complexity significantly. In initialization, a set of initial solutions are forwarded to encoding by the form of chromosome. They solution chromosomes are manipulated to produce child population solutions by evolutionary operator. Procedures in evolutionary operator to produce child population are called mutation and crossover. During each generation, effective population maintenance strategies are also adopted to keep track of the current non-dominated front.

2.2 Binary Encoding Strategy and Genetic Operator

Solutions population are encoded in the form of binary string by binary encoding strategy [22]. Mutation and crossover are performed on a binary string as illustrated in fig 21. One application of Binary-coded GAs is to conduct feature selection, chromosomes of solutions are encoded by

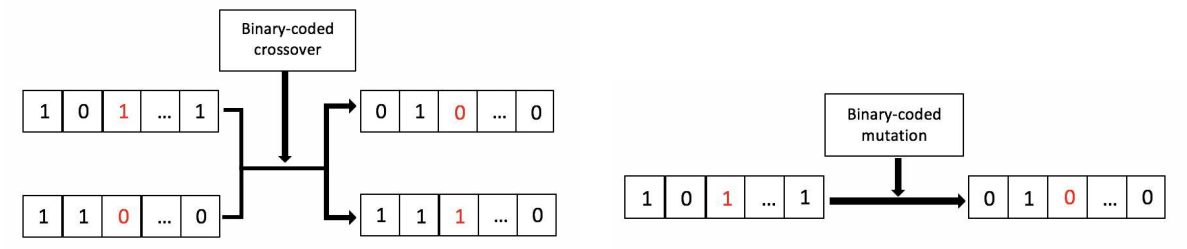

Figure 2.1 Conducting binary-coded crossover and mutation

binary string with length of total number of features. Bit '1' and '0' represent selected and ignored feature respectively, thus every solution denotes a subset of features. In Centroid-based encoding framework, positions of cluster center are encoded in the form of binary string. However, there are some weaknesses with binary-coded GAs in real-life implementation. The biggest one with binary-coded scheme is that Hamming cliff is likely to be triggered during searching. When searching around neighboring area of one particular solution, one might need to change many bits, i.e. 01111 $\longrightarrow$ 10000 which introduce lots of inconveniency to the gradual search in continuous searching space. This is inherently caused by uneven importance distribution among different schema in binary string. Moreover, it is difficult to achieve arbitrary precision since fixed string length limits the precision of solution and appropriate length of the string is often not known.

2.3 Real-coded Encoding Strategy

In order to solve these issues, real-coded GAs are adopted where population solutions are not required to convert into binary strings. Real-coded GAs are simple and straightforward. The key of constructing real-coded scheme is to imitate child population distribution results from binary coded scheme.

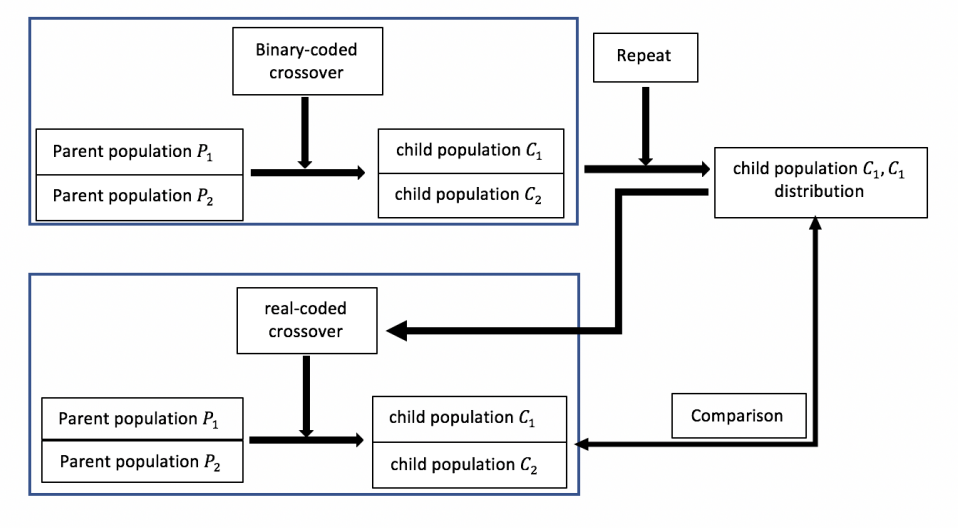

Figure 2.2 Real coded encoding strategy for Crossover

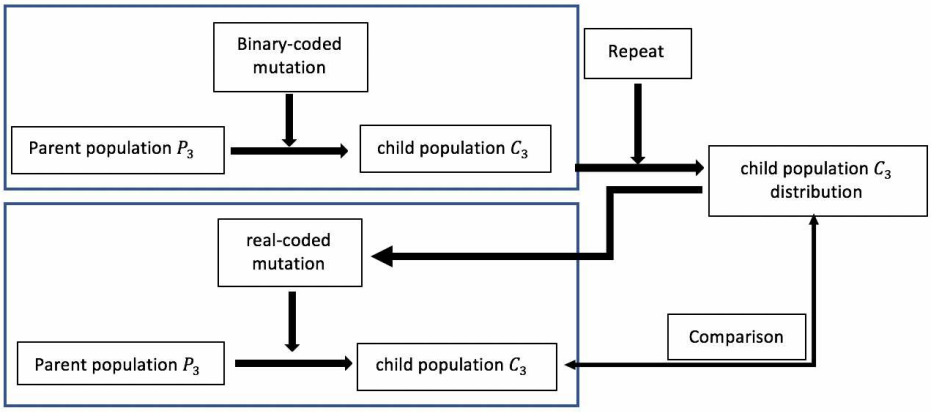

Figure 2.3 Real coded encoding strategy for Mutation

As shown above ,distribution of child population is produced by repeating binary- coded crossover and mutation process. These distributions are listed as reference in constructing real-coded scheme. Simulated binary crossover (SBC) is widely used and proposed by [23].

2.4 Simulation of SBC

Notice that probability of conducting crossover, number of bits changed during one crossover and position of crossover in binary scheme are all affecting distribution of child population distribution. Generally speaking, some properties of binary-coded crossover are important to estimate distribution. First of all, symmetry plays a significant role in distribution: $P_1 + P_2 = C_1 + C_2$, which

means we place equal importance on $P_1$and P_2. Secondly, P_1 and P_2 have the same probability P_c to be selected as a crossover point. Crossover in the lower bit and large bits results in small and large change respectively. In the end, child population are likely to be closed to parent population. SBC adopts probability density function of child distribution and simulates single-point crossover in binary-coded GAs (multiple-point crossover is superimposition of some single-point crossover). P_1 and P_2 are ready for crossover, $C_1 = \frac{P_1+P_2}{2} - \frac{\beta(P_2-P_1)}{2}$; $C_2 = \frac{P_1+P_2}{2} + \frac{\beta(P_2-P_1)}{2}$; β is functioned as a spreading factor. C_1 and C_2 are spread outward when $\beta > 1$, inwards when $\beta < 1$. Probability distribution of β should be similar to probability distribution of spread factor in binary-coded crossover. Probability density function(PDF) of β is defined as:

$$f(\beta) = \begin{cases} 0.5(n+1)\beta^n & \beta \leq 1 \\ 0.5(n+1)\frac{1}{\beta^{n+2}} & \beta > 1 \end{cases} \tag{2.1}$$

n =2 to 5 mostly. Another refined version is given by [23]:

$$C_1 = \frac{(1+\beta)P_1}{2} + \frac{(1-\beta)P_2}{2}; \quad C_2 = \frac{(1-\beta)P_1}{2} + \frac{(1+\beta)P_2}{2} \tag{2.2}$$

β is the distribution index. Where β can be generated each time by

$$\beta = \begin{cases} (2u)^{\frac{1}{\eta_c+1}} & u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}} & u > 0.5 \end{cases} \tag{2.3}$$

η_c is a variable that controls distribution concentration. As shown in fig 24 larger η_c tends to generate children closer to parents while small η_c allows children to be far from parents.

On the other hand, if η_c is comparatively small, distribution converges slowly. In extreme cases where large number of repetition is unreachable, distribution of child population is no longer guaranteed to have higher PDF around parent population. This is illustrated in fig[25]

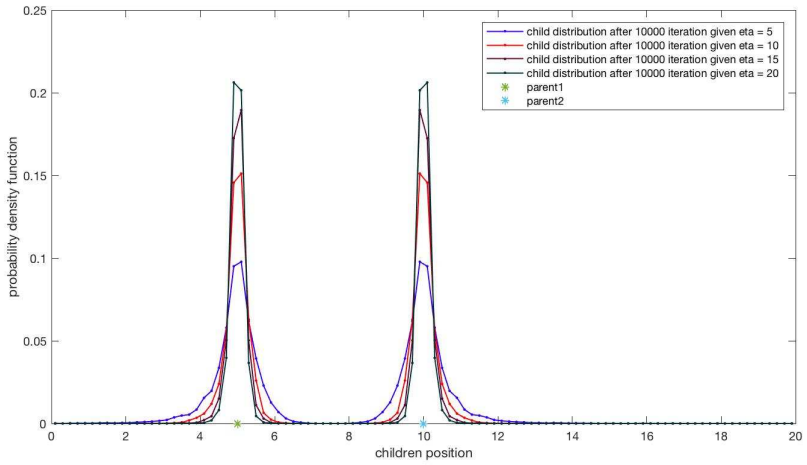

Figure 2.4 Child population distribution after 10000 iterations given $\eta_c = 5, 10, 15, 20$ for crossover

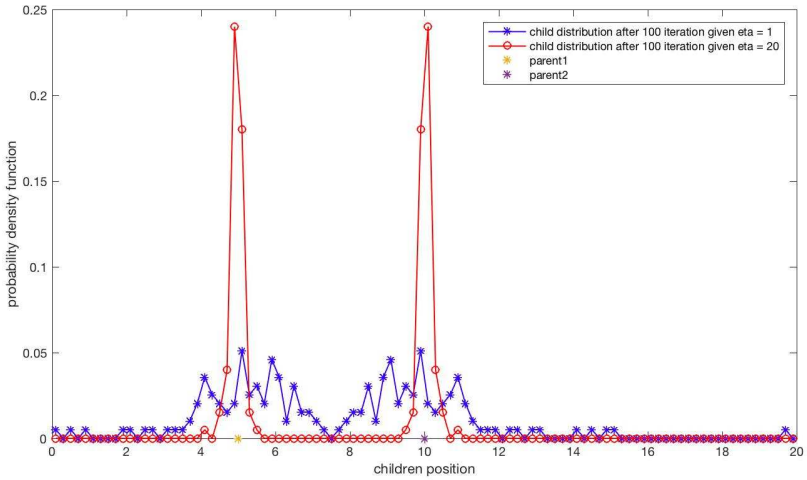

Figure 2.5 Child population distribution after 50 iterations given $\eta_c = 1$ and 20 for crossover respectively

We see that distribution of child population gradually converges to a smooth curve as number of simulation increasing.

Some valuable discoveries from simulation needs to be discussed. First of all, η_c can be considered as a spread factor in constructing crossover operator. Secondly, if η_c and the number of iteration is small, there will be some uncertainty with children distribution, therefore η_c should not be too small. On the other hand, children population are encouraged to spread out during early searching period. This is because parent solutions in this period are usually far away from

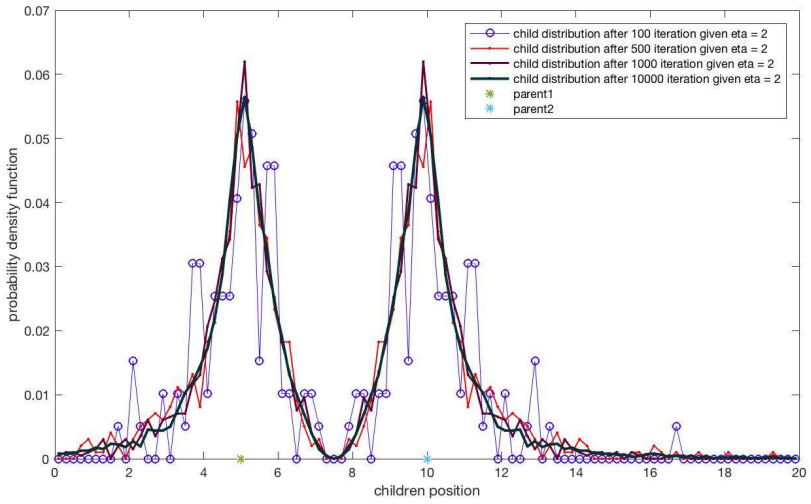

Figure 2.6 Child population distribution after 100, 500, 1000, 10000 iterations given $\eta_c = 2$ for crossover

Pareto optimal front and searching area should be large enough to reach out for better solution. Lastly, evolutionary operator should conduct searching in neighborhood area of parent population as iteration approaching termination. This is largely to that in later period, parent population solutions are assumed to locate in vicinity of Pareto optimal front. In summary, large η_c fails to satisfy our requirement in early searching stage while small η_c could results in wasteful searching. As a result, η_c should be a function of iteration with its value increasing as iteration going up.

2.5 Simulation of Real-coded Mutation

The design of mutation operator is quite similar. There are two types of mutation: uniform mutation and non-uniform mutation. The random mutation generates a solution randomly distributed within a vicinity of the original solution. Customarily, non-uniform mutation is adopted due to its preference on original solutions. There are several demands on selection of mutation operator. First of all, we wish mutated solutions are distributed towards parent population. Secondly, As the generation number increases, mutated solutions are generated closer to the original solution. Finally, it is obvious that all mutated solutions should be feasible. Mutation operator proposed by [20] is given as

$$C = P + 0.5\tau\Delta\left(1 - r^{(1-\frac{t}{t_{max}})^d}\right) \tag{2.4}$$

Δ is the maximum mutation step within constraint and τ is direction index where $\tau = 1$ and -1 indicate mutation towards upper and lower bound respectively. Mutation step is also affected by r which is randomly selected from 0 to 1. t and t_{max} denote number of mutation so far and the maximum number of mutation respectively. d is a variable that control concentration within

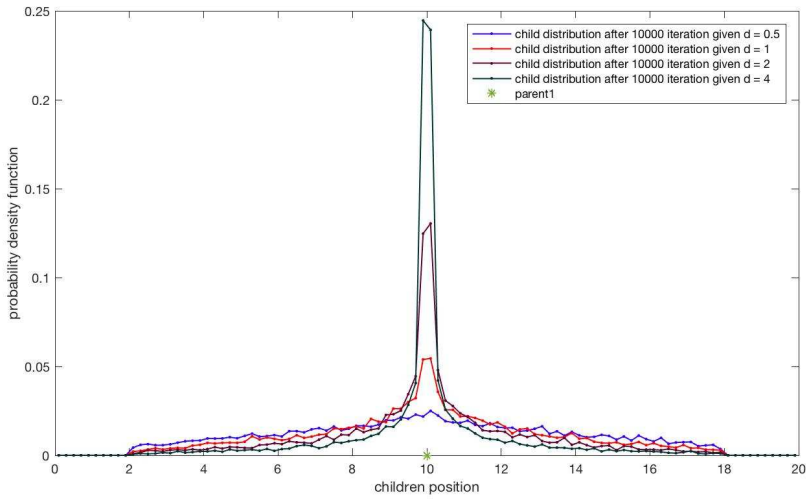

Figure 2.7 Child population distribution after 10000 iterations given different d for mutation

the child population distribution after mutation. Usually large d tends to generate children closer to parents while small d allows children to be far from parents. For convergence perspective, if d is comparatively small, distribution with insignificant number of trials β under test has some randomness. In other words, it is not guaranteed that child population are more likely to be near parent under insufficient simulation. On the other hand, if d is large enough, insufficient simulation won't be a problem

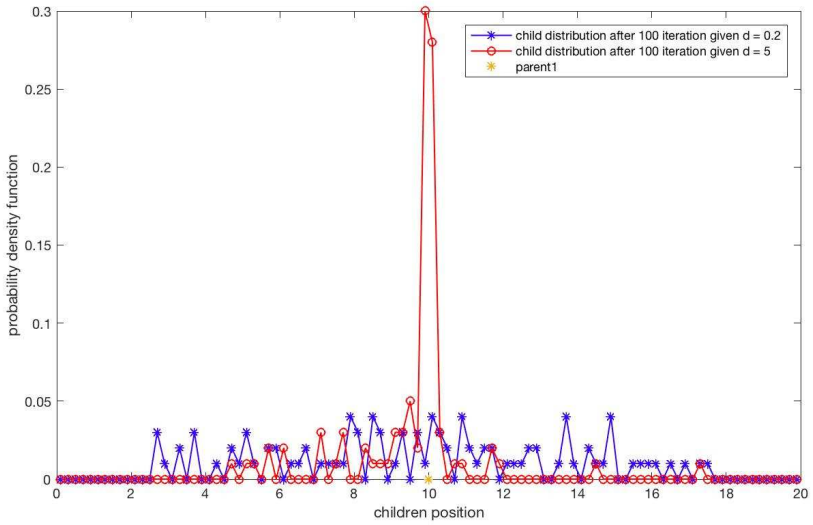

Figure 2.8 Child population distribution after 50 iterations given $d = 0.2$ and 5 for mutation respectively

With number of simulation increasing, distribution of child population gradually approaches to a smooth curve Some valuable discoveries from simulation also needs to be discussed. First of

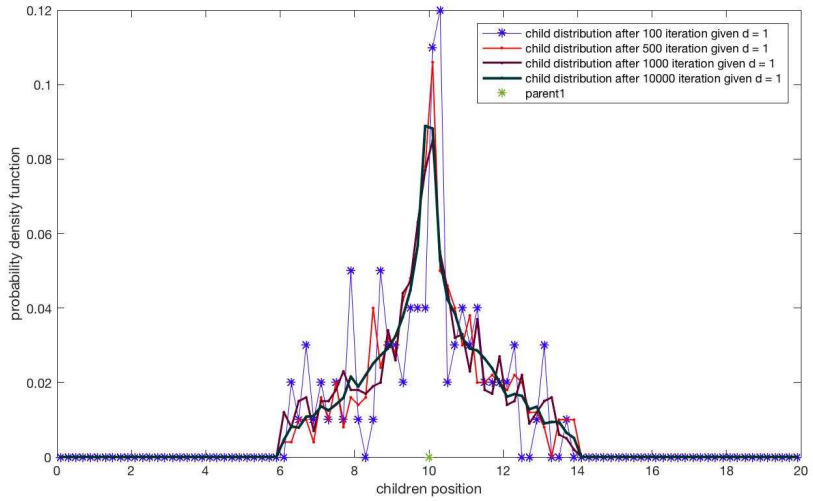

Figure 2.9 Child population distribution after 100, 500, 1000, 10000 iterations given $d = 1$ for mutation

all,d can be considered as a spread factor in constructing mutation operator. Secondly, if $d$and the number of iterations is small, there will be some uncertainty with children distribution, therefore d should not be too small. On the other hand, children population are encouraged to spread out during early searching period. This is because parent solutions in this period are usually far away

from Pareto optimal front and searching area should be large enough to reach out for better solution. Lastly, evolutionary operator should conduct searching in neighborhood area of parent population. This is largely to that in later period, parent population solutions are assumed to locate in vicinity of Pareto optimal front. In summary, large d fails to satisfy our requirement in early searching stage while small d could results in wasteful searching. As a result, d should be a function of iteration with its value increasing as iteration moving on.

2.6 Handling of Variable Length Chromosome in VMOGA

One proposed operator by [21] is capable of doing crossover when K is a variable. Assume parent solutions P_1 and P_2 encode K_1 and K_1 cluster centers. K_{max} and K_{min} denote upper and lower bound respectively. Considering data is always classified into at least 2 groups, $K_{min} = 2$. During iteration, crossover operator randomly selects an integer from $\max(2, K_1 + K_2 - K_{max})$, to $\min(K_{max}, K_1 + K_2 - 2)$ as variable length for child population. Procedure is illustrated in fig[210]:

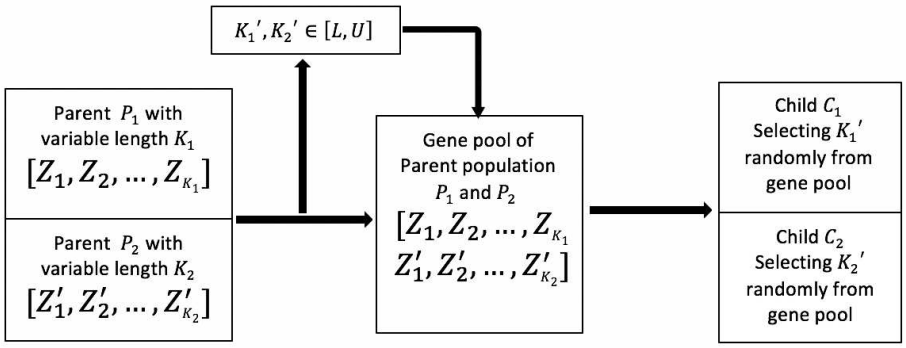

Figure 2.10 Genetic operator handle variable K by random selection from gene pool

Another crossover operation iproposed by [4] ensures exchanging solution information is conducted smoothly. Similarity, assume parent solutions P_1 and P_2 encode K_1 and K_2 cluster centers. λ_1 and λ_2 denote the location of crossover point in P_1 and P_2 respectively. λ_1 is generated by conducting random integer selection from 2 to K_{1m2}. λ_2 is generated by conducting random integer selection from $\underline{\lambda_2}$ to $\overline{\lambda_2}$. if $\overline{\lambda_2} \leq \underline{\lambda_2}$, $\overline{\lambda_2}$ is given by:

$$\underline{\lambda_2} = \min[2, \max(2 - (K_1 - \lambda_1))] \tag{2.5}$$

$$\overline{\lambda_2} = K_2 - \max[0, (2 - \lambda_1)] \tag{2.6}$$

Otherwise $\lambda_2 = 0$, This is illustrated in fig[211]

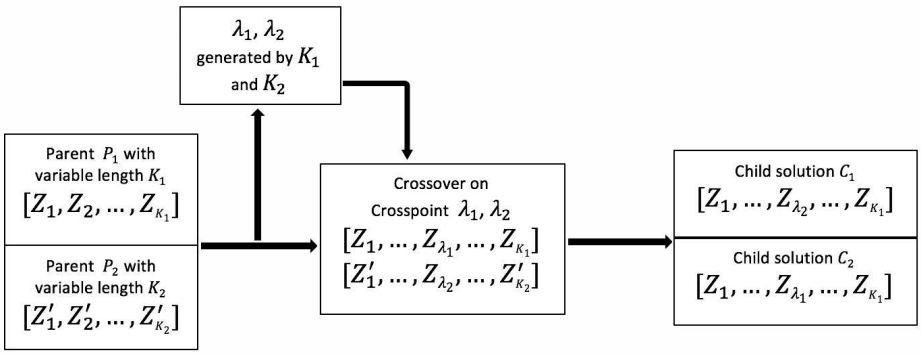

Figure 2.11 Genetic operator handle variable K by exchanging gene segment

Despite being easy to implement, genetic operator that handle variable K by random selection from gene pool do not follow binary crossover distribution. $K_1^{'}$ and $K_2^{'}$ are randomly selected from section with lowerbound and upperbound being $\max(2, K_1 + K_2 - K_{max})$ and $\min(K_{max}, K_1 + K_2 - 2)$ respectly, thus probability of child population to some extent will not be higher when it is getting close to parent population. Genetic operator that handle K by exchanging gene segment also has similar limitations. Moreover, it is unable to generate solutions with different length after initiation. In order to solve these issues, genetic operator in VMOGA is introduced. First of all, to achieve similar distribution when K is a variable, a similarity index is proposed to measure similarity between child and parent population.

$$S(P, C) = \begin{cases} \frac{|C|}{|P|} & \text{if adding genes} \\ \frac{|P \cap C|}{|P|} & \text{if losing genes} \end{cases} \tag{2.7}$$

Then, motivated by parameter setting in simulated crossover and mutation distribution, a spreading factor is generated from a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with μ determined by parent population and σ adopt the same strategy as η_c and d. In simulation, select operator conduct a random selection $\mathcal{N}(\mu, \sigma^2)$. $C_1 = P_1 \cup \text{select}(P_2)$ or $C_1 = \text{select}(P_1)$, each with 50 percent chance representing contraction and expansion. Results of simulation are illustrated in fig 2.12

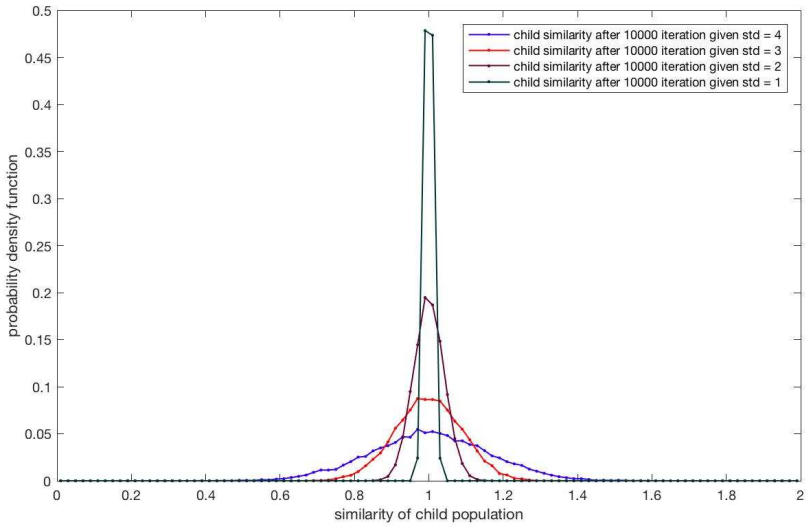

Figure 2.12 Child population distribution by VMOGA genetic operator given 10000 iterations for std = 1,2,3,4

CHAPTER 3

EVOLUTIONARY OPERATOR

3.1 Evolutionary strategy for Low-dimensional Space

Various strategies exist in the literature of searching in low-dimensional space. The most widely used technique is NSGA-II. NSGA-II conduct non-dominated sorting based on fitness value (objective value) and preserving diversity among solutions. The scheme to preserve diversity is to adopt crowding comparison operator. Generally, the operator prefers solutions with lower rank (sorting by non-domination) and if both solutions belong to the same rank, solutions located in a less crowded region are preferred. NSAG-III is a reference-point-based algorithm that adopt similar framework as NSGA-II

3.1.1 Non-dominated Sorting Algorithm

The concept of domination and Pareto front are given in introduction part. Here, we will see how these concepts applied in evaluating solutions population. Note that $\min_{x \in X} (f_1(x), f_2(x), ..., f_k(x))$, denotes $\overrightarrow{f} = (f_1, f_2, ..., f_k)$, $x_1, x_2, \ldots, x_N$ are N solutions. Basic idea is that we compare every $\overrightarrow{f(x_i)}$, with $\overrightarrow{f(x_1)}$,...,$\overrightarrow{f(x_{i-1})}$,$\overrightarrow{f(x_{i+1})}\ldots$, $\overrightarrow{f(x_N)}$. Comparison on each solution will bring computation complexity of $O(kN^2)$. In order to find the first non-dominated front, each solution is assigned two labels: domination count DC and dominated count DDC, representing the number of solution x_i dominated and the number of solution x_i being dominated. The only requirement for x_i to enter first front is $DDC(x_i) = 0$which means solution x_i is dominated by no one. Hence, no solution is being dominated in the first front. In order to find solutions in the next level, solutions of the first front are discounted temporarily and solutions of the first front among the remaining solutions enter marked as the second front. This is usually done by $\bigcup \lim_{i=1}^{N} DC(x_i)$ in practice. Procedure above is repeated until end of iteration and it is shown in fig **??**

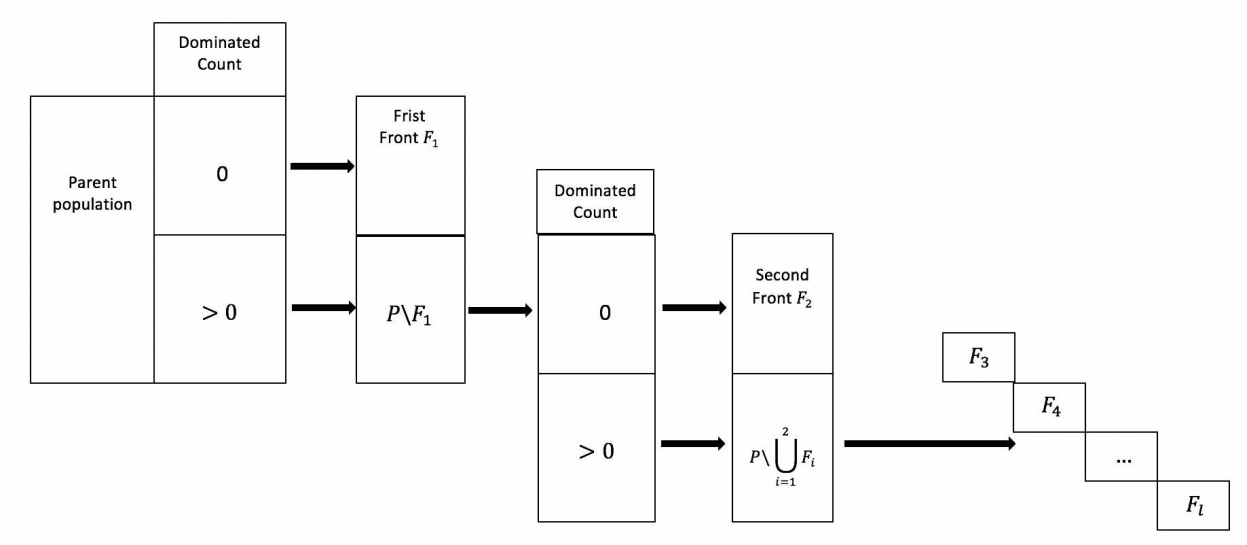

Figure 3.1 Framework of conducting non-dominated sorting and generating solution front

3.1.2 Diversity Preservation Algorithm

In NSGA-II, convergence of solution to the Pareto-optimal set is underlined. This requires evolutionary operator to maintain a good spread of solutions. In other words, our evolutionary operator must give an estimation of density of solutions surrounding a particular solution very quickly after non-domination sorting. Obviously, density estimation will be restricted in a certain area, i.e. neighborhood area of a given point. However, it is very difficult to determine how large this neighborhood should be, if it's too small, there might be no points fall in the area for most solutions being examined while it's computationally ineffective to construct such a large neighborhood. On the other hand, density estimation is fundamentally unreliable for high dimensional space. For relatively small dimensional space, crowding distance operator is proposed to handle this issue. Mostly, crowding distance requires sorting solutions with each objective function value in ascending order. For each objective function f_i, each boundary value $x_u = \{x : f_i(x) = \max(f_i(x))\}$or $x_l = \{x : f_i(x) = \min(f_i(x))\}$ is assigned an infinity value which emphasis significance of preserving spreading out solutions. For other solutions, overall distance is calculated as the sum height and width of cuboid. It is extremely useful in two-objective space since they only use one number to describe density. In other occasions where two or more features are required to depict density,

non-domination sorting rule could also be applied. In summary, diversity preserving operator gives every solution a "crowding estimation" DE and sort them by descending order, larger $DE(x_i)$ is preferred in diversity sorting which represent that solution x_i is located in a sparse area and preserving x_i will guide the searching toward a uniformly spread-out Pareto optimal front. Procedure for diversity preservation in NSGA-II is illustrated in fig[32]

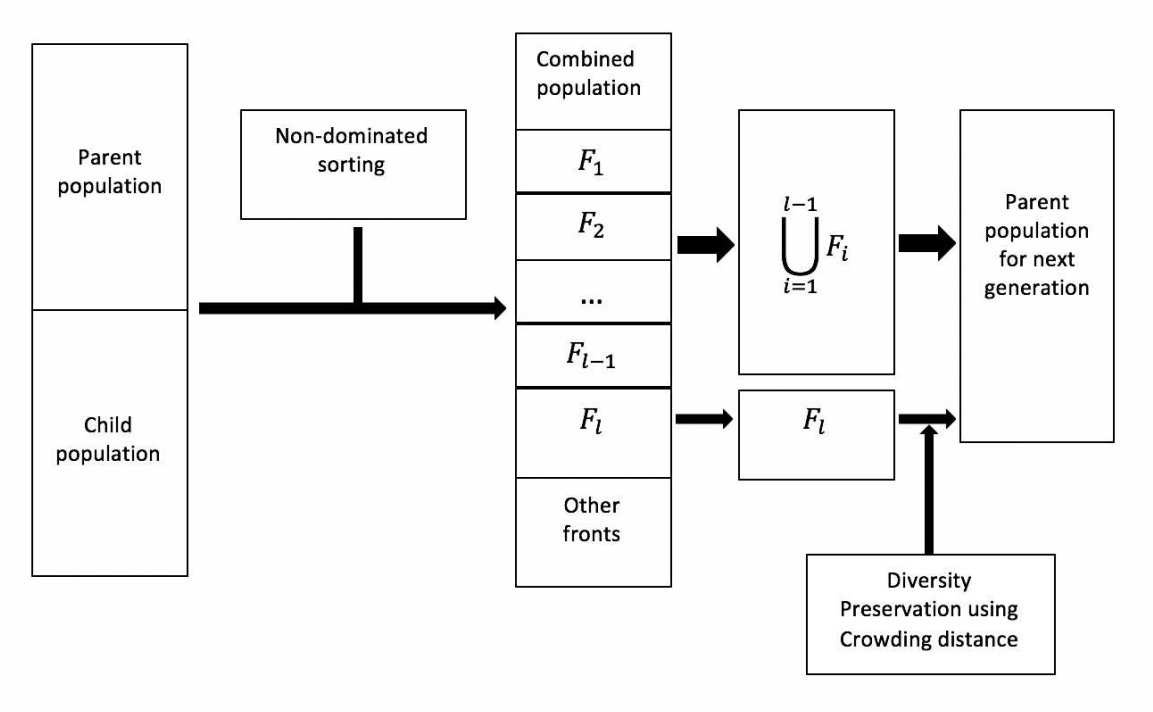

Figure 3.2 Framework of conducting diversity preservation in NSGA-II

3.1.3 Weakness

When dealing with real world problems, we often need to optimize more than four or more objectives. It is well known that with an increase number of objectives, an increasing proportion of solutions will be non-dominated. This is shown in fig[33]

Table 3.1 Main loop of NSGA-II

Step	Operation
Step1	Initial parent solutions P_1 with population number N are created with each population given a corresponding objective value.
Step2	Create offspring solutions Q_1 based on P_1 with population number N by genetic operator and corresponding objective values are assigned.
Step3	Combining $P_1 \bigcup Q_1 = R_1$. Sorting R_1 based on non-domination rule and generate $F_1, F_2, \ldots, F_i, \ldots$ Now we want to select N populations from R_1 in order to construct P_2, since l must exist satisfying $\lvert \bigcup_{i=1}^{l-1} F_i \rvert \leq N$ and $\lvert \bigcup_{i=1}^{I} F_i \rvert \geq N$ The first $l-1$ fronts $F_1, F_2, ..., F_{l-1}$ are assigned to P_2 and remaining $N - \lvert \bigcup_{i=1}^{l-1} F_i \rvert$ solutions will be selected from F_l by diversity preserving operator.
Step4	Sorting solutions in F_l by DE, $x = \{x : x \in F_l, DE(x) = \max(DE(x))\}$will be first selected, assigned to P_2, $F_l = F_l \backslash x$ and so on until $\lvert P_2 \rvert = N$.
Step5	Go to step2 until iteration is over.

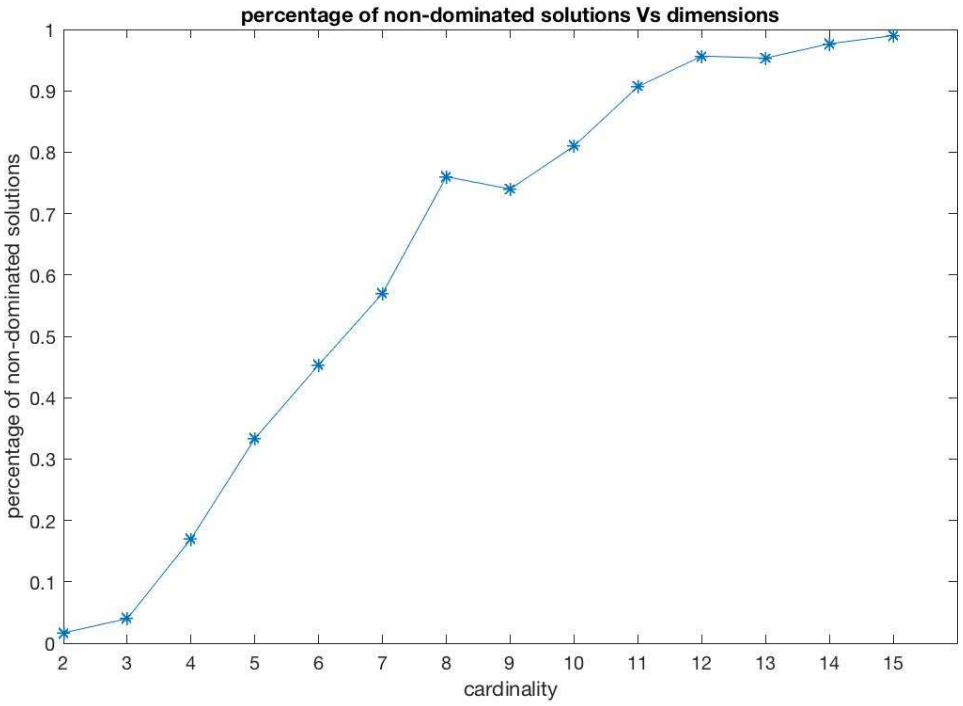

Figure 3.3 Non-dominated solutions proportion versus number of objectives

Figure[33] shows percentage of non-dominate solution versus cardinality For example, given $|P_t| = N$,$|Q_t| = N$ and $R_t = P_t \bigcup Q_t$, $|R_t| = 2N$. If proportion of non-dominated solution is more than 50 %, then $|F_1| > N$ which suggest that non-domination sorting is not working in population

selection. On the other hand, all the selecting pressure will be placed on diversity preserving operator. Recall that it is computationally expensive to calculate diversity in high-dimensional space, identification of neighbors which required crowded-comparison operator brings huge amount of calculation, approximation in diversity estimation to make computations faster may cause an unacceptable distribution of solutions at the end. To make things even worse, diversity preserving operator will sort N solutions in F_1 instead of solutions in F_l with population $N - |\bigcup_{i=1}^{l-1} F_i|$. In summary, traditional evolutionary operator leaves no space for in accommodating an adequate number of new solutions in the population and cause diversity preservation operator collapsing during iteration.

3.2 Evolutionary Strategy for High-dimensional Space

3.2.1 ε-Non-dominated Sorting Algorithm

In order to optimize many objectives simultaneously, a special domination rule is proposed by [24] by means of changing dominance rule. In minimization problem, ε-dominance can be expressed as: if x dominate y, then $\forall i \in 1, 2, ..., K, f_i(x) + \varepsilon_i \leq f_i(y)$ and $\exists i \in 1, 2, ...K, s.t \quad f_i(x) + \varepsilon_i < f_i(y)$. Another approach of conducting ε-dominance is quite similar: if x dominate y, $\forall i \in 1, 2, ..., K, f_i(x)(1 + \varepsilon_i) \leq f_i(y)$ (2) $\exists i \in 1, 2, ...K, \quad f_i(x)(1 + \varepsilon_i) < f_i(y)$ The concept of ε-Pareto optimal is given as, if $x^* \in X$ is ε-Pareto optimal, then $\forall x \in X$ if $\forall i \in 1, 2, ..., K \quad f_i(x^*) + \varepsilon_i \leq f_i(x), \exists i \in 1, 2, ...K, \quad f_i(x^*) + \varepsilon_i < f_i(x)$. Notice that ε_i must be predefined by some prior knowledge or universal rules. It has been proved the number of population in ε-Pareto optimal set is related to ε. Assume K objectives, if $\forall i \in 1, 2, ..., K, 1 \leq f_i(x) \leq M$

$$|F_\varepsilon| \leq \left(\frac{logM}{log(1+\varepsilon)}\right)^{K-1} \tag{3.1}$$

In fact, there are many different concepts of ε-Pareto optimality and ε-dominance exist in research literature. If we consider performance of ε non-dominated sorting on objective space, value of ε actually represents a relative tolerance or constraint on each objective. When $\varepsilon \to 0$, the ε-dominance place no extra constraint on solution population. As the value of ε increases, size of

$|F_\varepsilon|$ is decreasing over time. Idea behind this suggests a tradeoff between converging speed and diversity.

3.2.2 Reference Based Diversity Preservation Rule

In NSGA-III, a predefined set of reference point is adopted to maintain diversity. In this paper, we use Das and Dennis's [021] systematic approach that places points on a normalized hyper-plane. Imagine a mapping from objective space to a hyperplane, we now focus on geometric characteristic of hyper-plane. Reference points are uniformly distributed, and recombination population is mapped into this space. The number of reference points $\rho_j \in \rho$ can be computed by $\binom{d+P-1}{P}$ denote the number of objectives and divisions (along each objective) respectively. Similarity, we would like to see a solution getting higher chance of being selected when it is located in a less crowded region. Crowdedness here is measured by counting total number of solutions attached to the reference point. Reliable measurement must be designed in order to provide qualitative crowdedness description.

3.2.3 Density Estimation

Firstly, in order to make reasonable calculation, each objective function $f_i(x)$ is normalized: $f_i^{'}(x) = \frac{(f_i(x) - f_{min}(x))}{\max(f_i(x) - \min(f_i(x))}$, $x \in S_t$. Then, each reference point is associated with closest solutions. Here, distance between reference point and solutions are measured by projection on reference line. Reference line w is defined on the hyper-plane by joining reference point with the origin.

$$pd(x, w_j) = \left\| x - \frac{w_j^T x w_j}{\|w_j\|^2} \right\| \tag{3.2}$$

$pd(x, w_j)$ denotes perpendicular distance of solution x to reference line w_j. Then we calculate

$$\delta(x_i, w_j) = \begin{cases} 1 & \text{if} \quad pd(x_i, w_j) = \min\limits_{w \in W}(pd(x, w_j)) \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

Table 3.2 General procedure of Niche-Preservation

Step	Operation
Step1	Sorting combined solution population R_t (generation t) based on ε-nondomination rule. $F_1, F_2, \ldots, F_i, \ldots$ are generated.
Step2	Now we want to select N populations from R_1 in order to construct P_2 , since l must exist satisfying $\lvert \bigcup i = 1^{l-1} F_i \rvert \leq N$ and $\lvert \bigcup_{i=1}^{l} F_i \rvert \geq N$ The first $l-1$ fronts $F_1, F_2, \ldots, F_{l-1}$ are assigned to P_{t+1}
Step3	Now $N - \lvert \bigcup_{i=1}^{l-1} F_i \rvert$ solutions from F_l will be associated with reference points. $\theta(x) = \{\rho_m : \min_{\rho_j \in \rho} \rho_j\}$ are calculated for all solutions x in $\bigcup_{i=1}^{l} F_i$
Step4	Sorting solutions $x \in F_l$ by order of $\theta(x)$, $x \in F_l, \theta(x) = \min(\theta(x)\}$will be first selected, assigned to P_{t+1}, $F_l = F_l \backslash x$ and so on until $\lvert P_{t+1} \rvert = N$;

$\delta(x_i, w_j) = 1$ represent that solution x_i is associated with reference line w_j. The number of solutions associated with every reference line $w_j$is counted as

$$\rho_j = \sum_{x \in X} \delta(x, w_j) \tag{3.4}$$

ρ_j for reference line w_j provides valuable information to estimate solution density or crowdedness. Maximum diversity is then achieved by niche-preservation Operation.

3.2.4 Niche-Preservation Operation

We start with calculating

$$\theta(x) = (\rho_m | \rho_m = \min_{\rho_j \in \rho} \rho_j) \quad X = \{x : x \in X, \theta(x) = \min(\theta(x)\} \tag{3.5}$$

Least crowded reference lines are considered firstly. Solution population associated with these reference lines are selected randomly. Selection stops if the number of population reach N. Otherwise, after taking care of solutions associated with these reference lines, least crowded reference lines among the remaining are considered. General procedure is written as :

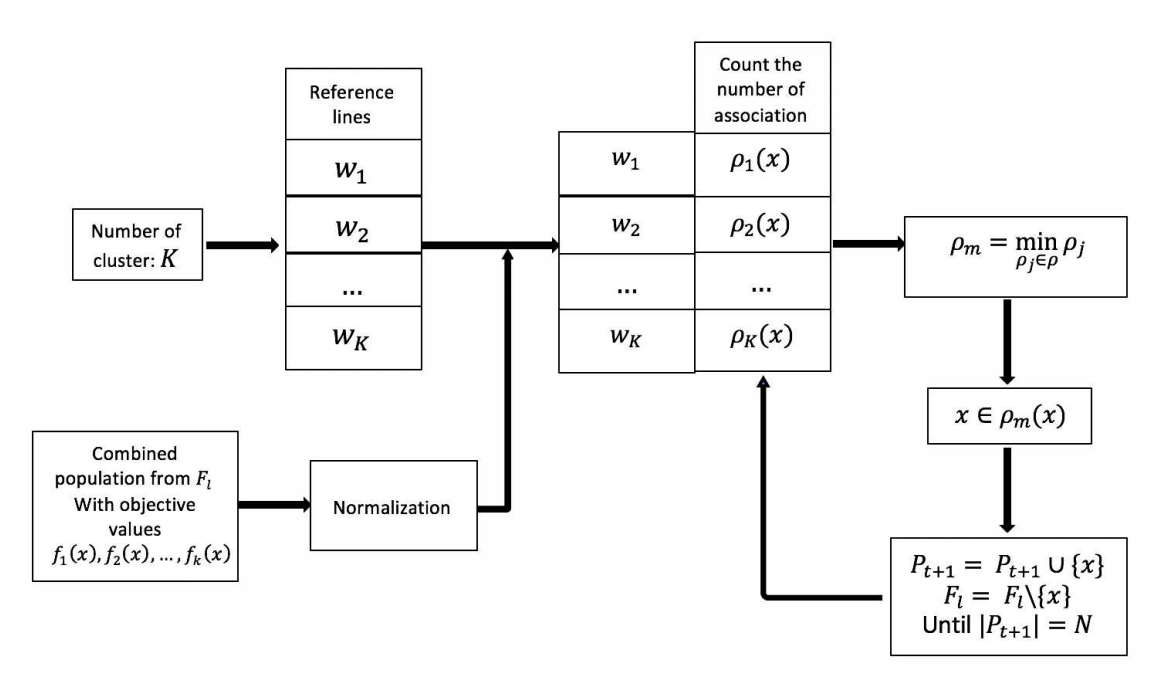

Figure 3.4 Framework of implementing niche preservation in NSGA-III

3.3 Evolutionary Operator for VMOGA

3.3.1 Tchebycheff Metric

In practice, parameter setting in ε-non-dominated sorting algorithm is difficult when prior knowledge about data to be clustered is un known. Usually, $\varepsilon_1 = ... = \varepsilon_d = \varepsilon$ with d representing the number of objectives. Solution that achieves superb performance in one objective while behave poorly on all the other objectives is still able to propagate to the next generation. For example, optimization problem is described as $\min_{x \in X}(f_1(x), f_2(x), \ldots, f_d(x))$, and $\exists x_0$, s.t $f_1(x_0) << f_1(x)$, $\forall x \in X \backslash x_0$ while $f_i(x_0) >> f_i(x), \forall x \in X \backslash x_0$ for $i = 2, 3, \ldots, d$. Even if ε is very large, still $(1 + \varepsilon) f_1(x_0) < f_1(x), \forall x \in X \backslash x_0$. Hence, x_0 always have great chance to be selected to first non-dominated front. Generally speaking, this issue can be alleviated (not extinguished) by adopting ε-dominance and setting large ε. However, large ε will significantly decrease the size of Pareto front, thus impairing diversity. Tchebycheff metric offers us inspiration in terms of removing bad

solution populoation. Proposed by [002], Tchebycheff metric is defined as

$$TCH(x, w, z^*) = \max_{i \in 1,\ldots,d} w_i \| f_i(x) - z_i^* \| \tag{3.6}$$

z^*denotes projection of idea point in each objective axis with weight vector $W = [w_1, \ldots, w_d]$ characterizing significance of each objective. In order to reject "unacceptable" solution population W could be set as $w_i = 1, i = 1, \ldots, d$. However, to determine the idea point is very hard. Sophistication of determining z^* also prevents us from using method. Furthermore, order of priority in rejection should be considered. Intuitively, solutions that deliver bad performance on $d - 1$ objectives ought to be removed, then $d - 2$ comes next, Finally, if rejection quota has not been used up, solutions with only one corrupted objective should be deleted. If equal importance is placed upon objectives, It is illustrated in analysis above that Tchebycheff metric basically ignore order of priority in rejection.

3.3.2 A Rejection Operator

Motivated by Tchebycheff metric and aggregating technique, a selection index V is proposed and defined as:

$$V(x) = \sum_{i=1}^{d} w_i f_i^*(x) \tag{3.7}$$

$f_i^*(x)$ denotes normalized objective value for solution x. Notice that V could take the advantage of Niche preservation process where normalization has done. ε-non-dominated sorting is still implemented in VMOGA after normalization of objective value. Here, ε is determined by previous experiment that illustrate relationship between non-dominated population proportion and value of ε. We need to ensure that

$$2N * P(\varepsilon) < N \tag{3.8}$$

As a supplement to non-dominated sorting, rejection operator has the rejecting rate of $\frac{\theta}{1+\theta}$%. Top $\frac{\theta}{1+\theta}$% population with largest $V(x)$ are rejected after Niche-Preservation Operation. In order to maintain population, a new l exist satisfying $|\bigcup_{i=1}^{l-1} F_i| \leq N(1+\theta)$ while $|\bigcup_{i=1}^{l} F_i| \geq N(1+\theta)$ .The number of $N(1+\theta) - |\bigcup_{i=1}^{l-1} F_i|$solutions are selected by reference-based diversity preservation

operator. Then $\frac{\theta}{1+\theta}$% of the Combined solutions with population $N(1+\theta)$are rejected. Evolutionary operator in VMOGA is illustrated in fig[35]

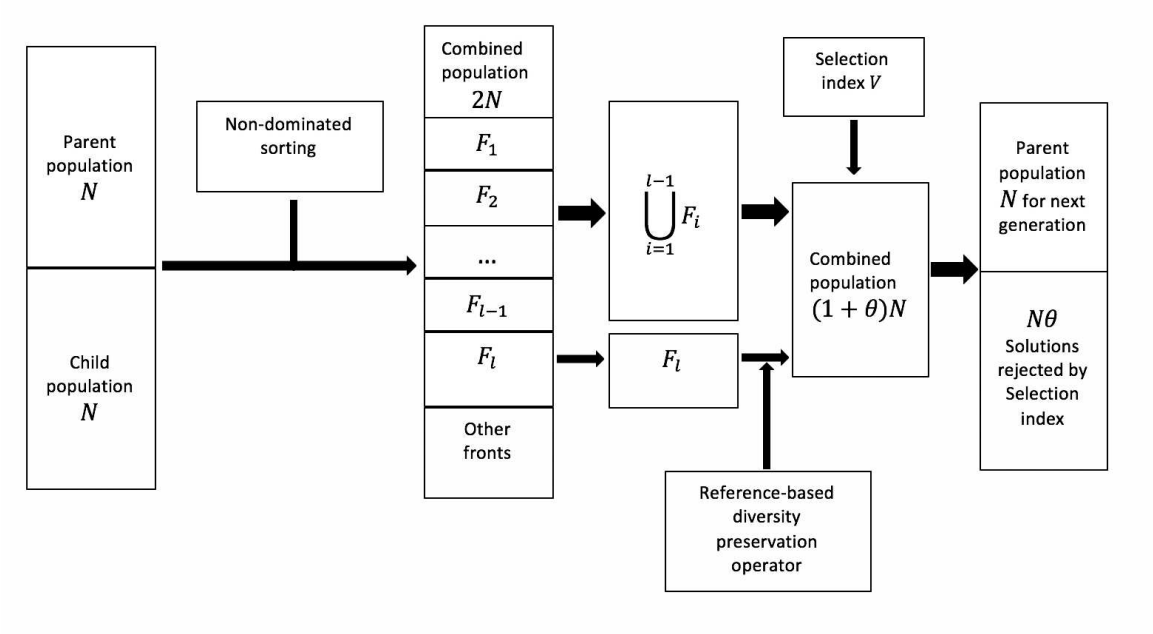

Figure 3.5 Evolutionary operator in VMOGA

CHAPTER 4

OBJECTIVE FUNCTIONS

4.1 Introduction

Objective functions that evaluate quality of clustering solutions are simultaneously optimized in multi-optimization framework. Early exploration of data clustering adopt validity index in evaluation of classification results since a solid connection exists between performance of clustering and validity index. When clustering performance is enhanced, validity index is optimized and vice versa. Applications of validity indexes as objective functions achieve satisfied results in previous studies [022]. Generally speaking, cluster prototype-based and cluster label-based approach are two types of objective functions exist in the literature. In 2005, cluster label-based encoding strategy is proposed by [023] and has been improved from then. However, it requires every data point being assigned a label, thus demands large computation. This issue can be tackled by Prototype-based approach. In this article, we adopt Prototype-based encoding strategy and introduce several frequently applied corresponding objectives. The importance of right choice of objective functions is also illustrated.

4.2 Distance Measure

Distance measure is the basic element in constructing measurement gauging similarity/ dissimilarity between two data points x_i and x_j. Suppose $X \in \mathrm{R}^{\mathrm{d}}$, distance between x_i and x_j is denoted as $D(x_i, x_j)$ which satisfying following properties:

- $D(x_i, x_j) \geq 0$ for all $x_i, x_j \in X$ and $D(x_i, x_j) = 0$ only if $i = j$
- $D(x_i, x_j) = D(x_j, x_i)$
- $D(x_i, x_j) \leq D(x_i, x_l) + D(x_l, x_j)$ for all $x_i, x_j \in X$

Table 4.1 Distance measurement

Name	Expression	characteristics
$D^{(1)}(x_i, x_j)$	$D^{(1)}(x_i, x_j) = \|x_{i,d} - x_{j,d}\|$, $x_{i,d}$and$x_{j,d}$ represent d^{th} component of $x_{i,d}$,$x_{j,d}$respectively that contain intensity information of image.	intensity information
$D^{(2)}(x_i, x_j)$	$D^{(2)}(x_i, x_j) = \sqrt{\sum_{l=1}^{l=d} \left(x_{i,l} - x_{j,l}\right)^2}$	Geographic and intensity information
$D^{(3)}(x_i, x_j)$	$D^{(3)}(x_i, x_j) = \sqrt{\sum_{l=l_1,l_2,\ldots,l_k} \left(x_{i,l} - x_{j,l}\right)^2}$.	$l_1^{th}, \ldots, l_k^{th}$ feature of data space

Euclidean distance measure is the most commonly used one in research, here, we categorize them into three groups based on characteristics representation.

4.3 Membership Function

Membership function is vital in constructing objectives. In this paper, we wish to build up a framework containing a set of universally applicable objective functions. Cluster validity indices according to [022], are often selected as the objective function and they are heavily depend on calculation of membership value. Membership function can be written as a matrix U with its value reflecting belongingness of data point to corresponding cluster. It is also equivalent to partition matrix. In image segmentation problems, fuzzy membership value is calculated by:

$$u_{i,j} = \frac{1}{\sum \lim_{i=1}^{K} \left(\frac{D(z_k, x_j)}{D(z_i, x_j)}\right)^{\frac{2}{m-1}}}, 1 \le i \le K; 1 \le j \le n; \tag{4.1}$$

$u_{k,j} \in [0, 1]$ and simple calculation shows that:

$$\sum \lim_{k=1}^{K} u_{k,j} \left(\sum \lim_{k=1}^{K} D(z_k, x_j)^{\frac{2}{m-1}} \right) \left(\sum \lim_{k=1}^{K} \sum \lim_{j=1}^{n} u_{k,j} \right) = 1; \quad \sum \lim_{k=1}^{K} \sum \lim_{j=1}^{n} u_{k,j} = n \tag{4.2}$$

Equation[4.2] proves that $u_{k,j}$ is reasonable to reflect belongness. $D(z_i, x_k)$ denotes distance between cluster centerz_i and pattern x_k. It should be pointed out that $D(z_i, x_k)$ could be

$D^{(1)}(x_i, x_j)$or $D^{(2)}(x_i, x_j)$ in following studies determined by practical demands. In image segmentations, $D^{(1)}(x_i, x_j)$ and $D^{(2)}(x_i, x_j)$ are adopted when intensity distribution of pixels and geographic information are considered respectively. Fuzzy exponential m is usually set as 2, leading $\frac{2}{m-1} = 2$. During implementation, centers encoded in a chromosome are first extracted (usually given by a random generation) and are forwarded to objective function.

4.4 Refined Membership Function

In practice, $u_{k,j}$ is likely to be affected by noise and artifacts. In theory, we hope pixel or pattern gets bigger chance of being classified into the same cluster as neighborhood pixels. As a result, refined membership function is proposed by [024]. By evaluating belongingness of surrounding area, neighborhood information is incorporated into refined membership value. Another advantage of refined membership function is that ambiguity between adjacent patterns (i.e. overlapping) will be considerably reduced. In order to construct refined member function, some preparations are needed. First of all, diameter of neighboring region is predefined (usually set as 3 or 5) and crisp membership value is calculated via

$$g_{k,j} = \begin{cases} 1 & if u_{k,j} \geq u_{l,j}, l = 1, 2, ..., K \\ 0 & otherwise \end{cases} \tag{4.3}$$

Secondly, estimation on partition results of neighboring region is given by

$$h_{k,j} = \sum_{j \in neighborhood} g_{k,j} \tag{4.4}$$

$h_{k,j}$ is defined as the summation of crisp membership value $g_{k,j}$ in a given neighboring region. In the end, refined membership function can be constructed by product of $h_{k,j}$ and $u_{k,j}$ with normalization.

$$u^*_{k,j} = \frac{u_{k,j} h_{k,j}}{\sum\limits_{k=1}^{K} u_{k,j} h_{k,j}}, \quad 1 \leq i \leq K, 1 \leq j \leq n \tag{4.5}$$

Similarly, $u^*_{k,j} \in [0, 1]$ and simple calculation shows that:

$$\sum \lim_{k=1}^{K} u^*_{k,j} \left(\sum_{k=1}^{K} D(z_k, x_j)^{\frac{2}{m-1}} \right) \left(\sum_{k=1}^{K} \sum \lim_{j=1}^{n} u^*_{k,j} \right) = 1; \quad \sum_{k=1}^{K} \sum_{j=1}^{n} u^*_{k,j} = n \tag{4.6}$$

4.5 Objective Functions

4.5.1 Fuzzy C-means (FCM)

FCM [025][026] is a widely used objective in partition matrix evolution. In fact, single-objective optimization once took FCM as optimizing objective because of its capability of upgrading cluster center every iteration. In this paper, J_m is only considered as one of objective functions and it's given by

$$FCM = \sum_{j=1}^{n} \sum_{k=1}^{K} u_{k,j}^{m} D^2(z_k, x_j) \tag{4.7}$$

Where n is the number of data point or pattern (number of pixels in image equivalently), $D(z_k, x_j)$denotes Euclidean distance between data points x_j and the center z_k. Notice that J_m summates variance over all clusters. Lower value of J_m indicate better compactness and each data point is classified to the cluster that achieves the largest membership value. It's quite intuitive that J_m decrease as number of cluster K increasing and it can be proved that J_m takes minimum value when $K = n$

4.5.2 Xie-Beni Index (XB)

Xie-Beni index [027] a function of the ratio of the total variation σ to the minimum separation sep of the clusters. Total variation σ is given by

$$\sigma = \sum_{j=1}^{n} \sum_{k=1}^{K} u_{k,j}^{m} D^2(z_k, x_j) \tag{4.8}$$

which is the same as J_m and minimum separation $\min sep(V) = \min_{i \neq j} \{\|z_i - z_j\|^2\}$, is a measure of the worst case scenario of separation in clustering. XB is written as

$$XB(U, V; X) = \frac{\sigma(U, V; X)}{n \times min\_sep(V)} \tag{4.9}$$

4.5.3 Overall Cluster Deviation (DEV)

$$Dev = \sum_{k=1}^{K} \sum_{x_j \in C_k} D^2(z_k, x_j) \tag{4.10}$$

When maximum membership for each data point is assigned 1 and others by 0,

$$J_m = \sum k = 1^K \sum_{x_j \in C_k} D^2(z_k, x_j) \tag{4.11}$$

In fact, Dev can be viewed as crisp version of J_m. During iteration ambiguity on the boundary is eliminated since summation in every iteration is proceed right after crisp classification. Dev must be minimized in order to obtain compact clusters. This objective is practically the same as the objective of K-means clustering.

4.5.4 Fuzzy Separation (FESP)

Fuzzy separation is given by [019]

$$S = \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} \mu_{i,j}^{m} D^2(z_i, x_j) \tag{4.12}$$

Membership degree follow similar equation to measure belongingness of each z_i to z_j, $i \neq j$.

$$\mu_{i,j} = \frac{1}{\sum\limits_{l=1, l \neq j}^{K} \left(\frac{D(z_j, z_i)}{D(z_j, z_l)} \right)^{\frac{2}{m-1}}}, j \neq i \tag{4.13}$$

It should be maximized in order to obtain well separated clusters.

4.5.5 Global Separation (SEP)

Global separation [022] is introduced as supplement criteria in estimating separation and it can be considered as the crisp version of fuzzy separation. Maximizing sep(V) encourage clusters to be separated from each other, on the other hand, distance between cluster centers should be large enough to avoid overlapping.

$$sep(V) = \frac{2}{K(K-1)} \sum \lim_{i=1}^{K} \sum_{i \neq j} D^2(z_i, z_j) \tag{4.14}$$

4.5.6 Average Between Group Sum of Squares (ABGSS)

ABGSS is proposed by [028] which also measure cluster separation and compute the average distance of the cluster centers from the centroid of the dataset as follows:

$$ABGSS = \frac{\sum \lim_{i=1}^{K} n_i D^2(z_i, \overline{z})}{K} \tag{4.15}$$

Here, $\overline{z}$ represent the center of the whole dataset and n_i is the number of data points having maximum membership value corresponding to z_i. $ABGSS$ must be maximized in order to obtain well-separated clusters.

4.5.7 Intra-cluster Entropy (H)

Intra-cluster entropy is brought up in terms of improving clustering homogeneity [029]. Average degree of similarity between each cluster center and data points is given by

$$g(z_k) = \frac{1}{n} \sum \lim_{j=1}^{n} \left[0.5 + \frac{CO(z_k, x_j)}{2} \right] \tag{4.16}$$

Value of $g(z_k)$ reflect global probability of grouping all data points to C_k represented by center z_k. cosine distance $CO(x_i, x_j)$ is written as $CO(x_i, x_j) = \frac{\overrightarrow{x_i} \cdot \overrightarrow{x_j}}{\|\overrightarrow{x_i}\| \|\overrightarrow{x_j}\|}$ Intra-cluster entropy of partition C_i is given by

$$H(Z_i) = -[g(z_i) \log_2 g(z_i) + (1 - g(z_i)) \log_2 (1 - g(z_i))] \tag{4.17}$$

In the end, general Intra-cluster entropy is defined as

$$H = \sum_{k=1}^{K} \left[1 - H(Z_i) g(z_i) \right]^{\frac{1}{d}} \tag{4.18}$$

d is the dimension of dataset. High intra-cluster entropy is desirable because low entropy from each cluster enhance homogeneity of our results.

4.5.8 Davies-Bouldin Index (DB)

The Davies-Bouldin index [030] is a function of the ratio of the sum of within-cluster scatter to between-cluster scatter, scatter of cluster is given as:

$$S_i = \frac{\sum\limits_{x_j \in Z_i} D^2(z_i, x_j)}{|Z_j|} \tag{4.19}$$

Single intra-cluster homogeneity and compactness can be estimated by

$$R_k = \max_{j \neq k} \left\{ \frac{S_k + S_j}{D^2(z_k, z_j)} \right\} \tag{4.20}$$

Overall homogeneity and compactness is then given by:

$$DB = \frac{\sum \lim_{k=1}^{K} R_k}{K} \tag{4.21}$$

Davies-Bouldin index calculate the ratio of within cluster scatter to intra cluster separation. For a good partition result, inter cluster separation as well as intra cluster homogeneity and compactness should be high. This affirms the idea that no cluster has to be similar to another, and hence the best clustering scheme essentially minimizes the Davies–Bouldin index.

4.5.9 Connectedness

Inspired by concept of fuzzy adjacency relation from [01], For any pair of pixel (x, y), the affinity function $\mu_k(x, y)$indicates the local hanging togetherness and it is written as:

$$\mu_k(x, y) = \mu_\alpha(x, y) g(\mu_\psi(x, y), \mu_\phi(x, y)) \tag{4.22}$$

$\mu_\alpha(x, y)$ represents adjacency relation which is defined as:

$$\mu_\alpha(x, y) = \begin{cases} 1 & if \|x - y\| \leq 1 \\ 0 & otherwise \end{cases} \tag{4.23}$$

g should be a monotonically non-decreasing function and two components of $\mu_\psi(x, y)$ and $\mu_\phi(x, y)$ represent object-feature based and homogeneity-based function. In our case, calculation of

$g(\mu_\psi(x, y), \mu_\phi(x, y))$ can be simplified into

$$g(\mu_\psi(x, y), \mu_\phi(x, y)) = \begin{cases} 1 & x, y \in C_k \\ 0 & otherwise \end{cases} \tag{4.24}$$

The shortest path between two points x,y is denotes as $d_{short}(x, y)$, which is measured along the relative neighborhood graph. First of all, all possible paths connecting x and y are found, assume $x_1, x_2, \ldots, x_k$ are intermediate point forming path $C(x_1, x_2, \ldots, x_k)$ which satisfy

$$\mu_k(x_k, x_{k+1}) = 1 \tag{4.25}$$

Set $|C(x_1, x_2, \ldots, x_k)| = k$, $d_{short}(x, y) = \min_{c \in C} |c|$. Otherwise, if there is no path between (x, y), set $d_{short}(x, y) = \inf$. For a given cluster C_k, we calculate the number of connected components and denote it as $L(C_k)$

$$Conn = \max_{k \in 1,2,\ldots,K} L(C_k) \tag{4.26}$$

Cluster validity index is minimized during optimization.

CHAPTER 5

EVALUATION OF OBJECTIVES

As mentioned before, there is no single objective function that works well for all kinds of dataset. Moreover, performance of multi-optimization algorithm is also likely be harmed by objectives that provide insufficient coverage of data features. Therefore, selection of objective is quite important. An increasing number of objectives generally enhance optimizing performance while adding computational complexity. Multi-objective optimization is performed on a number of, often conflicting objectives. The degree of conflict between two objectives increases if optimization of one objective violate the assumption of the other. One example is to optimize XB and FCM: FCM is strongly correlated with global variance while XB index is a combination of global variance and minimum separation as numerator and denominator respectively. Separation between closest clusters are considered as the worst-case scenario here, the value of which increases as clusters are well separated from each other, thus increase global variance. It is shown in [015], minimizing XB index and FCM simultaneously yields good results. It turns out to be much more difficult to give similar analysis on other objective functions depending limited knowledge. Therefore, information theory is introduced to solve this issue.

5.1 Multivariate Mutual Information

5.1.1 Mutual Information Among Two Variables

Generally speaking, mutual information [031] measures amount of information about one variable given the knowledge of another variable and it is mostly defined on two variables. In probability theory, mutual information gives a quantitative measurement on the amount of information obtained about one variable through another. Our problem here requires the knowledge of interaction between more than two variables. Firstly, in two-objective cases, assume X and Y have distribution

$P_X(x)$,$P_Y(y)$ respectively. Entropy is given by

$$H(X) = -\sum_{x \in X} P_X(x) \log P_X(x) \tag{5.1}$$

$$H(Y) = -\sum_{y \in Y} P_Y(y) \log \left(P_Y(y)\right) \tag{5.2}$$

Then joint entropy is written as

$$H(X, Y) = -\sum_{x \in X, y \in Y} P_{XY}(x, y) \log \left(P_{XY}(x, y)\right) \tag{5.3}$$

Mutual information is defined in terms of individual entropy and joint entropy:

$$I(X; Y) = H(x) + H(Y) - H(X, Y) \tag{5.4}$$

Now, we want to expand to many-objective situation, the idea underlying this is dimension reduction.

5.1.2 Mutual Information Among Multi Variables

For three variables, U, V, Y. If V has no effect on U and Ythen $I_V(U; Y) = I(U; Y)$, $I_v(U; Y) = I(U; Y)$. On the other hand, if V affect the relationship between $U$and Y, we could eliminate V by taking a weighted sum (on the probability of occurrence of the particular value of V) of the mutual information between U and Y for each value of V.

$$I_V(U; Y) = \sum_{v \in V} P_V(v) I(U; Y|V = v) = I(U; Y|V) \tag{5.5}$$

$$I(U; Y|V) = I(U, V; Y) - I(V; Y) \tag{5.6}$$

since

$$I(U, V; Y) = H(U, V) + H(Y) - H(U, V, Y); I(U; Y) = H(U) + H(Y) - H(U, Y) \tag{5.7}$$

We can defer that

$$I_V(U; Y) = H(U, V) - H(U, V, Y) - H(V) + H(V, Y) \tag{5.8}$$

A simple subtraction between$I_V(U; Y)$ and$I(U; Y)$ can be written as

$$I(U; Y) - I_V(U; Y) = H(U) + H(V) + H(Y) - (H(U, V) + H(V, Y) + H(U, V)) + H(U, V, Y) \tag{5.9}$$

This is also called mutual interaction between V,U and Y.Analysis above presume that probability distribution of each variable is known. Now, we can easily extend mutual information calculation to a universal case:

$$I(X_1; X_2) = H(X_1) - H(X_1|X_2); H(X_1|X_2) = I(X_1|X_2) \tag{5.10}$$

So, $I(X_1; X_2) = I(X_1) - I(X_1|X_2)$, with knowledge above

$$I(X_1; X_2; X_3) = I(X_1; X_2) - I(X_1; X_2|X_3) \tag{5.11}$$

And

$$I(X_1; X_2; \ldots; X_N) = I(X_1; X_2; \ldots; X_{N-1}) - I(X_1; X_2; \ldots; X_{N-1}|X_N) \tag{5.12}$$

These equations can be expressed in terms of entropy

$$I(X_1; X_2; ...; X_N) = (H(X_1) + H(X_2) + ... + H(X_N)) - ... + (-1)^{N-1} H(X_1; X_2; ...; X_N) \tag{5.13}$$

When processing only two variables,I is always nonnegative, if $I = 0$, $X$and $Y$are independent and knowing X does not give any information about Y and $I > 0$ implies information is shared by $X$and Y, the larger$I$ is, more information is shared by X and Y. The inefficiency with multi-variate mutual information is that it could be either positive or negative. For instance, consider case that U and Y are independent of each other and V is a variable only dependent on U, then $I(U; V; Y)$ is negative.

5.2 Total Correlation

Total correlation is the amount of information shared among the variables in the set. In information theory, total correlation is one of the several generalizations of mutual information. For a given set of N random variables $\{X_1, X_2, \ldots, X_N\}$, total correlation is defined as Kullback-Leibler divergence from the joint distribution $p(X_1, X_2, \ldots, X_N)$ to the independent distribution of $p(X_1)p(X_2)\ldots p(X_N)$:[032]

$$C(X_1, X_2, \ldots, X_N) = D_{KL}[p(X_1, X_2, \ldots, X_N)|p(X_1)p(X_2)\ldots p(X_N)] \tag{5.14}$$

By reducing divergence to simpler difference of entropies:

$$C(X_1, X_2, \ldots, X_N) = \sum_{i=1}^{N} H(X_{i)} - H(X_1, X_2, \ldots, X_N) \tag{5.15}$$

which can be also written as

$$C(X_1, X_2, \ldots, X_N) = \sum_{i,j} I(X_i; X_j) + \sum_{i,j,k} I(X_i; X_j; X_k) + \ldots + I(X_1; X_2; \ldots; X_N) \tag{5.16}$$

CHAPTER 6

IMPLEMENTATION

6.1 Minor Specification

6.1.1 Handling of Infinity Solution in Objective Space

It's not rare in practice when prior knowledge is inaccessible and number of cluster in implementation is unknown. An infinite value in objective space is possible, for instance, when some clusters are empty, Z_i, hence $|Z_i| = 0$. Recall that within-scatter of cluster is given as:

$$S_i = \frac{\sum\limits_{x_j \in Z_i} D^2(z_i, x_j)}{|Z_j|} \tag{6.1}$$

S_i goes to infinity. Notice that normalization will be affected by doing calculation $\frac{\infty - C}{\infty}$. This is unlikely to generate a reasonable result. In our implementation, solutions with infinite objective are detected immediately and deleted.

6.1.2 Handling of Solutions with Same Objectives or Diversity

Solutions with the same objectives or diversity are always selected randomly until solution population reach threshold N.

6.1.3 Handling of Solutions with Same Membership on Two Clusters

Assume that data pointx_j has equivalent maximum membership for cluster Z_1 and Z_2: $u_{1,j} = u_{2,j}$ due to fuzzy clustering. This often implied that x_j is on the boundary of Z_1 and Z_2. In order to take advantage of fuzzy clustering, data pointswith equivalent degree of belongingness are assigned to more than one clusters. Some objectives, for instance, deviation requires calculating $Dev = \sum\limits_{k=1}^{K} \sum\limits_{x_j \in C_k} D^2(z_k, x_j)$. In VMOGA, x_j is assigned to both $Z_1$and Z_2.

6.2 Parameter and Experiment Setting

6.2.1 Initialization and Objective Selection

Table 6.1 Initialization parameter

Initialization Parameter	value
Number of generations	50
Population size	100
Parent solution population	100
Child solution population	100
Upper bound on number of clusters	30
Lower bound on number of clusters	2

Six different methods are selected to construct comparison Fuzzy c-means (FCM) is a method of clustering which allows one pixel to belong to two or more clusters. FCM is proposed by [9] and frequently applied in pattern recognition. It is based on minimization of the FCM objective function

$$FCM = \sum_{j=1}^{n} \sum_{k=1}^{K} u_{k,j}^{m} D^2(z_k, x_j) \tag{6.2}$$

Partition is carried out through an iterative optimization of FCM shown above, with the update of membership $u_{i,j}$ and the cluster centers z_k ,

$$u_{i,j} = \frac{1}{\sum_{i=1}^{K} \left(\frac{D(z_k, x_j)}{D(z_i, x_j)} \right)^{\frac{2}{m-1}}}, 1 \leq i \leq K; 1 \leq j \leq n; \tag{6.3}$$

$$z_k = \frac{\sum_{j=1}^{n} u_{k,j}^{m} x_j}{\sum_{j=1}^{n} u_{k,j}^{m}} \tag{6.4}$$

NSGA-II is one of the most widely adopted two-objective optimization genetic algorithm. In our experiment, MOVGA and NSGA-II optimize the same objectives (FCM and XB) while the number of cluster$K$ is fixed for NSGA-II and set as a variable for MOVGA. Similarily, NSGA-III and VMOGA-1 both optimize FCM,XB, $1/FSEP$, DB, $1/H$ while K is fixed for NSGA-III and set as a variable for VMOGA-1. In VMOGA-2, another objective function $Conn$ in terms of evaluating

connectedness is incorporated into our framework. However, introduction of $Conn$ disrupts the balance of previous objective set. Number of cluster K, which is set as a variable in VMOGA, is adopted to address this issue. SO (Single optimization of $FCM + XB + 1/FSEP + DB + 1/H$) is taken as a supplement criteria in our comparison, with corresponding weight value to each objective set as one. Recall that weight value or priority for each objective function is needed before implementing aggregating methods. Optimizing SO in our experiment is to exclude the possibility that equal weights happen to be the right answer since in this case rejection operator invalidate the rest part of evolutionary operator.

Table 6.2 Choice of objectives for FCM, SO,NSGA-II, MOVGA, NSGA-III and VMOGA

Methods	Choice of objectives					
	J_m	XB	1/FuzzySep	DB	1/H	Conn
FCM	√					
SO	√	√	√	√	√	√
NSGA-II	√		√			
MOVGA	√		√			
NSGA-III	√	√	√	√	√	√
VMOGA	√	√	√	√	√	√

[1] SO optimize normalized $J_m + XB + 1/FuzzySep + DB + 1/H + Conn$

6.2.2 Parameter Encoding and Genetic Operator

Methods with predefined number of clusters K (NSGA-II, NSGA-III, SO) adopt simulated binary crossover with $\eta_c = 10$ and mutation operator with $d = 10$ with Crossover probability and mutation probability equal 1 and 0.2 respectively. In VMOGA, crossover operation is determined by normal distribution $\mathcal{N}(\mu, \sigma^2)$, μ is the length of the parent solution chromosome, $\sigma = 4 - 3\left(\frac{t}{t_{max}}\right)^{\frac{1}{2}}$, t denotes iteration insofar, $t_{max} = 200$ as total number of generation. Mutation operator parameter

$d = 0.1 + 9.9\left(\frac{t}{t_{max}}\right)^{\frac{1}{2}}$; Moreover, rejection rate $\theta = 0.1$ for VMOGA. In refence-based sorting algorithm, provision is set as 4.

6.2.3 Obtaining Final Solution

Notice that, a set of non-dominated solutions are generated by multi-optimization algorithm and none of these solutions can be improved in one objective without degrading another. Thus, we need do some tradeoff or use a selection scheme to choose one particular solution. Index I [6] is efficient as a supplement validity index to measure the goodness of cluster results. It is given by

$$I(K) = \frac{1}{K} \times \frac{E_1}{E_K} \times D_K \tag{6.5}$$

K denotes the number of clusters. Observed from equation, Index I is composed of $\frac{1}{K}$,$\frac{E_1}{E_K}$ and D_K. $E_1 = \sum_{j=1}^{n} D(z_1, x_j)$;$E_K = \sum_{k=1}^{K} \sum_{x_j \in Z_k} D(z_k, x_j)$. E_1 is taken as constant and z_1 represent global center of dataset. Recall that E_K has same expression as Dev which needs to be minimized in order to enhance compactness of clustering results. $D_K = \max_{i,j} D(z_i, z_j)$; it denotes the largest distance between two cluster centers and larger D_K indicate better separation. In summary, index$I$ is to be maximized. During implementation, we select solution from the last generation with largest I.

6.3 Experiment on MRI Data

6.3.1 Introduction of MRI Data

Image Segmentation plays an important role in medical image analysis. Automatic segmentation of MRI brain images into different classes is very important in clinical study and neurological pathology. However, image segmentation has always been challenging for MRI images since these images are noisy and imprecise in nature. One important aspect of evaluating segmenting results is to calculate regional volume. In real-life application, regional volume calculations often bring very useful diagnostic information. Among them, the quantization of gray and white matter volumes may

be of major interest in neurodegenerative disorders such as Alzheimer disease, movement disorders such as Parkinson or Parkinson related syndrome, white matter in metabolic or inflammatory disease, congenital brain malformations, perinatal brain damage, or in post-traumatic syndrome. Normal brain image data provided by [22] is a great source to evaluate efficiency of proposed algorithm. Here, ground truth table is available for these images. There are ten classes present in Normal brain image with each class given an integer between 0 and 9 representing its content.

Table 6.3 Truth table for MRI brain image

Class	Back gound	CSF	Grey Matter	White Matter	Fat	Muscle/ skin	Skin	Skull	Glial Matte	Connective
Label	0	1	2	3	4	5	6	7	8	9

A result comparison here is established with MOVGA proposed by [4] and evaluated by adjusted rand index(ARI) [25]

6.3.2 Adjusted Rand Index(ARI)

Clustering quality is mostly evaluated by external measurement. The Rand index(RI) takes two partitions as the input (one of which is the correct solution from truth table). The number of pairwise co-assignments of data items between the two partitions is counted as RI. Adjusted Rand index (ARI) [25] additionally introduces a statistically induced normalization in order to yield values close to zero for random partitions. Suppose true partition is available and denotes as $T = \{T_1, T_2, \ldots, T_r\}$ for MRI data $X = \{X_1, X_2, \ldots, X_N\}$, our segmentation results are written as $C = \{C_1, C_2, \ldots, C_s\}$. Overlapping between T and C can be summarized in a contingency table illustrated in table[64]

Table 6.4 Contingency table

	C_1	C_2	...	C_s	Sums
T_1	n_{11}	n_{12}	...	n_{1s}	a_1
T_2	n_{21}	n_{22}	...	n_{2s}	a_2
...	...	...	...	...	...
T_r	n_{r1}	n_{r2}	...	n_{rs}	a_r
Sums	b_1	b_2	...	b_s	

Each element in contingency table denotes the number of objects belonging to $T_i \bigcap C_j$ as $n_{ij} = |T_i \bigcap C_j|$, and $\sum_{i=1}^{r} a_i = \sum_{i=1}^{s} b_i = N$

$$ARI = \frac{\sum_{i,j} \binom{n_{i,j}}{2} - \frac{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right]}{\binom{N}{2}}}{\frac{1}{2}\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2} - \frac{\left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right]}{\binom{N}{2}}} \tag{6.6}$$

ARI=1 when clustering results perfectly match true partition T and approximate to 0 for random classification.

6.3.3 Evaluation of Segmentation Results

Algorithms have been applied on the images corresponding to the Z planes Z10, Z72, Z108, Z120 and Z140.Results are presented in Fig[61:65]. It appears that proposed VMOGA has identified different homogeneous regions very well

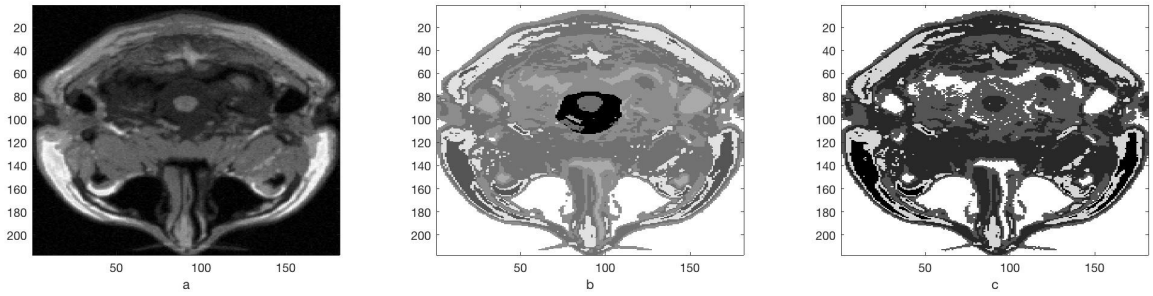

Figure 6.1 (a)Original MRI image in Z10 plane (b)Ground truth table (c)Corresponding segmented image produced by VMOGA clustering

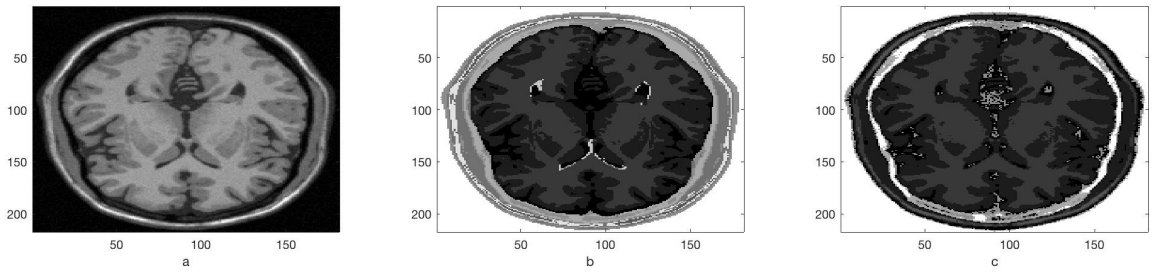

Figure 6.2 (a)Original MRI image in Z72 plane (b)Ground truth table (c)Corresponding segmented image produced by VMOGA clustering

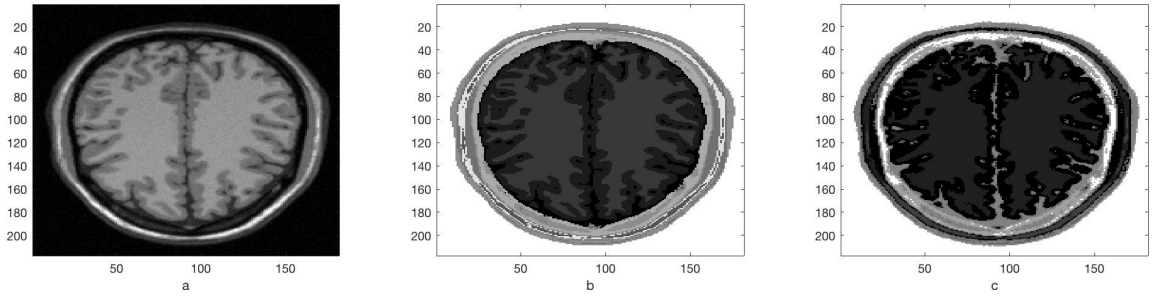

Figure 6.3 (a)Original MRI image in Z108 plane (b)Ground truth table (c)Corresponding segmented image produced by VMOGA clustering

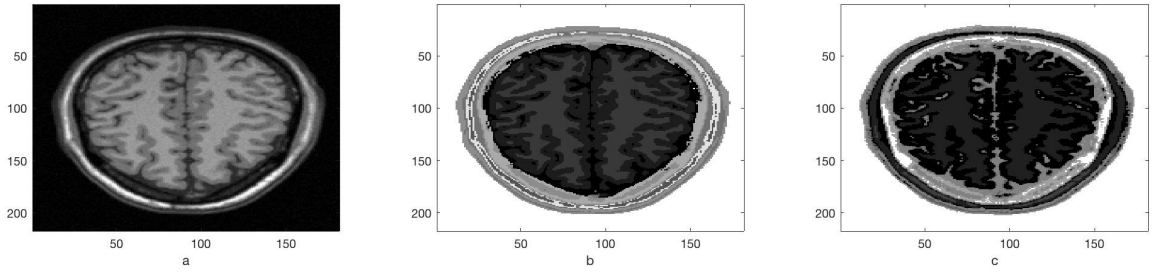

Figure 6.4 (a)Original MRI image in Z120 plane (b)Ground truth table (c)Corresponding segmented image produced by VMOGA clustering

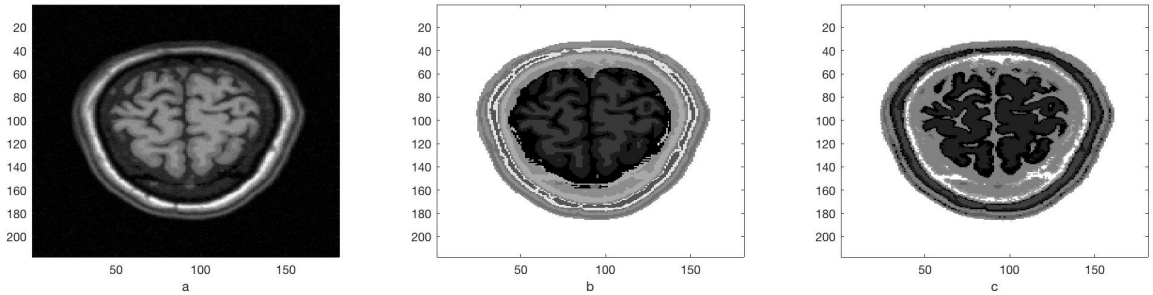

Figure 6.5 (a)Original MRI image in Z140 plane (b)Ground truth table (c)Corresponding segmented image produced by VMOGA clustering

Results of ARI score are illustrated in table [65]

Table 6.5 Results for Z10, Z72, Z108, Z120 and Z140 planes

	FCM	SO	NSGA-II	NSGA-III	MOVGA	VMOGA
Z10	0.661	0.509	0.722	0.786	0.744	0.801
Z72	0.604	0.450	0.727	0.725	0.664	0.719
Z108	0.719	0.557	0.732	0.791	0.767	0.808
Z120	0.698	0.386	0.761	0.810	0.758	0.825
Z140	0.627	0.591	0.816	0.886	0.827	0.893

It is evident in table[65] that proposed VMOGA generally outperforms other algorithm. Meanwhile, the presence of VMOGA and NSGA-III results illustrate that more objectives provide extra coverage of data characteristics, hence enhance clustering performance. Comparison of FCM, NSGA-II and NSGA-III further illustrate the importance of optimizing many objectives simultaneously. Results of single-objective optimization is also presented in the table. Recall that a rejection operator V is adopted in VMOGA to remove bad solutions. It is also neccessary to eliminate the possibility that best results could be selected by index V individually. The single optimization of V here displays that rejection operator is incapable of selecting good solutions.

6.3.4 Detection of White and Gray Matter

Further application on MRI images segmentation results is going to be estimation of white and gray matter. Gray matters are important reference in diagnosing neurodegenerative disorders such as Alzheimer disease, Parkinson or Parkinson related syndrome while white matter is important for metabolic or inflammatory disease, congenital brain malformations, perinatal brain damage, or in post-traumatic syndrome. Here detection accuracy (AAC) is defined as

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{6.7}$$

TP, FP, FN, TN can be interpreted as True positive: white/gray matter correctly identified as white/gray matter, False positive: white/gray matter incorrectly identified as non-white/gray matter, True negative: non-white/gray matter correctly identified as non-white/gray matter, False negative:

non-white matter incorrectly identified as white matter respectively.Moreover, VA(Volume prediction accuracy) is proposed to examine accuracy of white or gray matter prediction. VA is given by

$$VA = \frac{|P - T|}{|T|} \tag{6.8}$$

Where T and P denotes true volume and predicted volume by segmentation respectively Comparisons are made on Gray and White Matter segmentation with respect to truth label and VMOGA results. It is illustrated in fig[66-68]

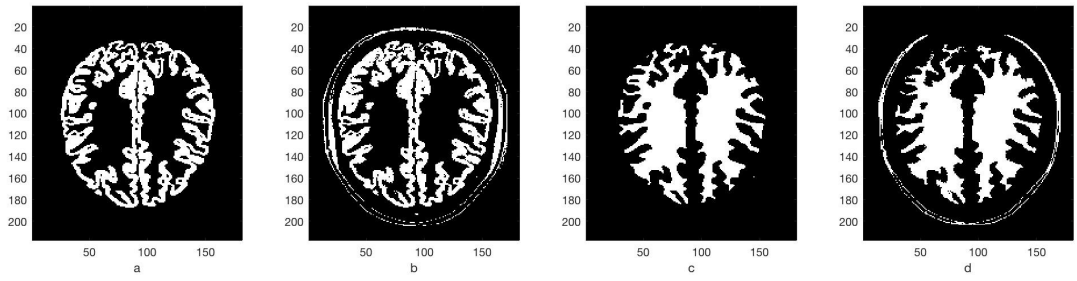

Figure 6.6 (a)Gray Matter Ground truth (b)Gray Matter segmentation produced by VMOGA on Z108 plane (a)White Matter Ground truth (b)White Matter segmentation produced by VMOGA on Z108 plane

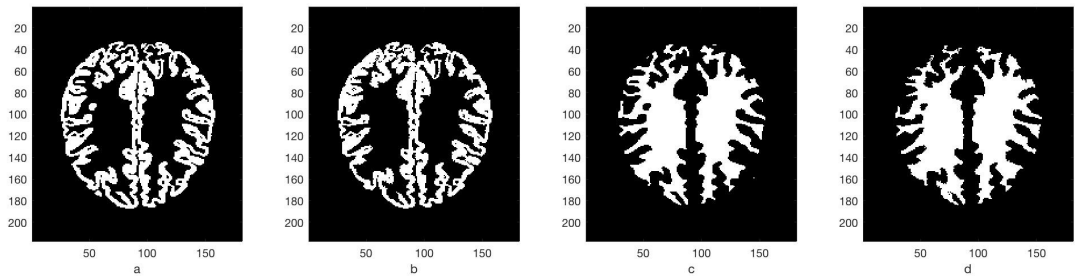

Figure 6.7 (a)Gray Matter Ground truth (b)Gray Matter Extraction produced by VMOGA+Extraction on Z108 plane (a)White Matter Ground truth (b)White Matter segmentation produced by VMOGA-based extraction on Z108 plane

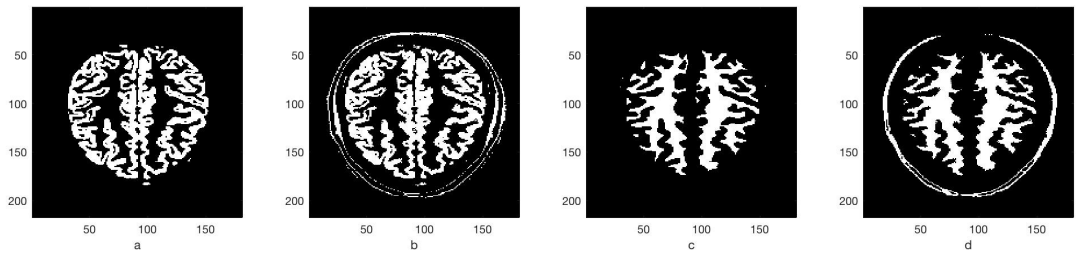

Figure 6.8 (a)Gray Matter Ground truth (b)Gray Matter segmentation produced by VMOGA on Z120 plane (a)White Matter Ground truth (b)White Matter segmentation produced by VMOGA on Z120 plane

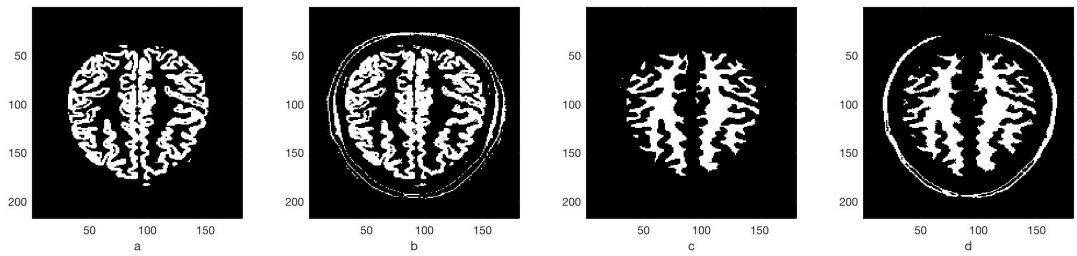

Figure 6.9 (a)Gray Matter Ground truth (b)Gray Matter Extraction produced by VMOGA+Extraction on Z120 plane (a)White Matter Ground truth (b)White Matter segmentation produced by VMOGA-based extraction on Z120 plane

Table 6.6 Gray and White Matter extraction results for normal brain images for Z72, Z80, Z100, Z108 and Z120 planes.

MRI plane	Index	Performance evaluation						
		FCM	SO	NSGA-II	NSGA-III	MOVGA	VMOGA	Extraction
Z72	Gray Acc	0.86	0.692	0.873	0.881	0.892	0.899	0.959
	Gray VA	39%	46%	33.8%	35.2%	31.9%	26.2%	3.15%
	White Acc	0.873	0.777	0.912	0.904	0.915	0.924	0.978
	White VA	47%	59.5%	40.4%	25.1%	29.6%	17.7%	3.3%
Z80	Gray Acc	0.852	0.791	0.839	0.860	0.881	0.884	0.970
	Gray VA	50%	64%	40.8%	46.5%	39.2%	25.3%	4.68%
	White Acc	0.901	0.832	0.942	0.933	0.939	0.948	0.971
	White VA	41%	60%	27.9%	13.8%	22.2%	16.5%	3.85%
Z100	Gray Acc	0.906	0.821	0.906	0.91	0.904	0.895	0.982
	Gray VA	35%	66%	40.8%	46.5%	39.2%	26.2%	2.34%
	White Acc	0.876	0.707	0.912	0.934	0.925	0.941	0.987
	White VA	40%	37%	30.9%	15.8%	29.6%	15.1%	1.09%
Z108	White Acc	0.893	0.802	0.932	0.884	0.915	0.951	0.977
	White VA	40%	49.5%	30.9%	15.8%	29.6%	17.7%	3.21%
	Gray Acc	0.895	0.844	0.811	0.948	0.864	0.947	0.985
	Gray VA	38%	77.2%	33.1%	38.4%	22.6%	19.9%	0.53%
Z120	White Acc	0.907	0.772	0.883	0.952	0.910	0.961	0.978
	White VA	40%	106%	34.9%	35.8%	41%	26%	1.75%
	Gray Acc	0.929	0.821	0.881	0.93	0.901	0.944	0.984
	Gray VA	32%	83%	17.2%	16%	19.4%	12.9%	0.21%

Figure [66-69] shows the clustered gray and white matter using VMOGA technique. Visually it provides a similar clustering structure as that provided corresponding truth table. However, VMOGA fails to identify skin, skin/muscle matter which disturbs measurement accuracy. There-

fore, connectedness-based extraction is adopted as a supplementary procedure in quantization of Gray and White Matter and corresponding segmentation result of Z100 and Z120 plane is shown in Fig [66-69].It is illustrated in the table[66] that proposed VMOGA generally achieve higher accuracy when extracting White and Gray Matter. Meanwhile, the presence of VMOGA and NSGA-III results illustrate that increasing number of objectives provide growing coverage of data characteristics. Comparison of FCM, NSGA-II and NSGA-III further illustrate the importance of optimizing many objectives simultaneously. Furthermore, it is shown in the table [66] that evident discrepancies exist in the quantization of Gray and White Matter from VMOGA. Superiority of proposed Extraction (results from VMOGA plus connectedness based extraction) has been demonstrated which significantly reduce the error of detection and volume prediction accuracy.

6.4 Implementation on Fatigue Area Detection

The proposed image processing method is validated on NDE data obtained from inspection of delamination in glass fiber reinforced polymer (GFRP) specimens subjected to Mode I fatigue tests. A GFRP specimen is periodically imaged from its healthy state to damaged state using optical transmission scan [26]. OTS has been previously used for tracking impact damage growth in GFRP specimens [27]. In the experiment, sample is illuminated by the laser source and power transmitted through the sample is recorded by photodetector to examine extent of delamination in the GFRP sample. At first, the critical displacement where the specimen cracks are recorded by introducing monotonic loading to five similar samples. The process is then repeated for all the specimens and the average critical displacement is computed. Fatigue loading is then conducted on a new sample under cyclic loading at 5 cycles/sec and displacement ratio of 0.1. OTS images of mode 1 GFRP sample subjected to 8 rounds of cyclic loading are presented in Figure [610]:

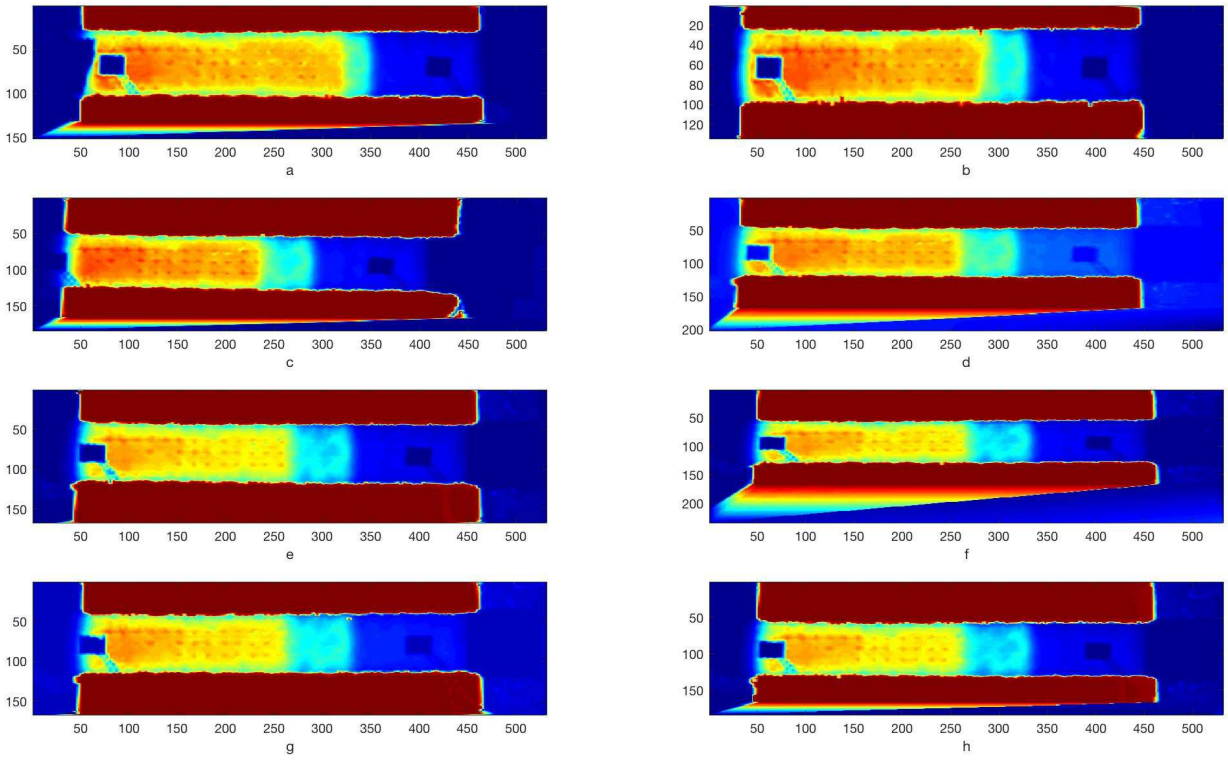

Figure 6.10 Healthy sample being subjected to Mode 1 cyclic loading after (a) 20K cycles (b) 40K cycles (c) 60K cycles (d) 80K cycles (e) 100K cycles (f) 120K cycles (g) 140K cycles (h) 160 cycles

From OTS images, extent of delamination can be observed as the region between end of Teflon and the beginning of healthy part of the sample. It is shown that delamination inside specimen grows slowly as the number of load circles increasing. The piezoelectric sensors attached to the GFRP sample are marked as reference points and are used to calculate the physical area of delamination from d_{pix}. During preprocessing, location of the two PZT sensors are identified and the pixel distance between their inner edges is recorded as l_{pix} .Additionally, upper and lower edges of the sample and its pixel width is recorded as w_{pix} . The cutting results are presented in Fig[611]

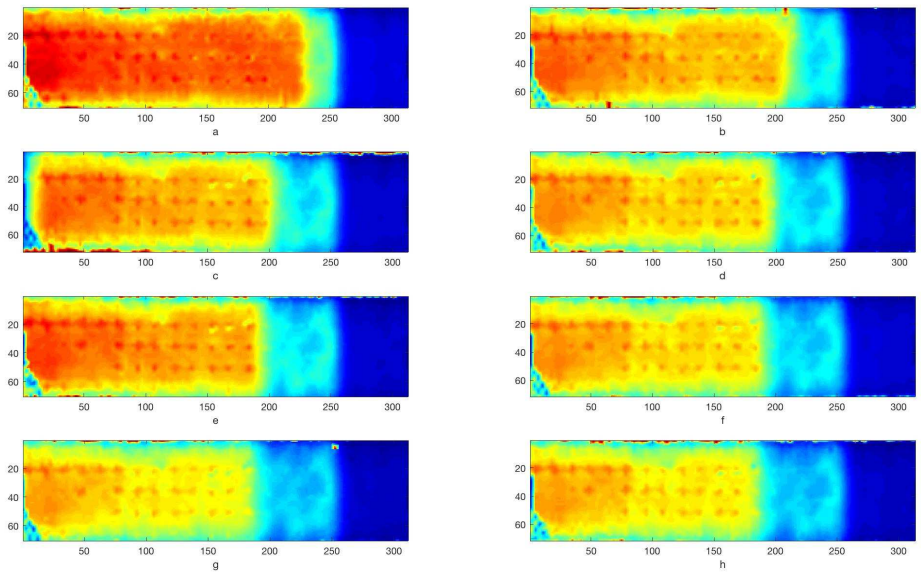

Figure 6.11 PZT cutting image of Healthy sample being subjected to Mode 1 cyclic loading after (a) 20K cycles (b) 40K cycles (c) 60K cycles (d) 80K cycles (e) 100K cycles (f) 120K cycles (g) 140K cycles (h) 160K cycles

Area of delamination from the OTS image is originally computed using image processing algorithm implemented in MATLAB. The delaminated area is identified using segmentation via fast marching method (FMM) [02] whose corresponding threshold for extracting fatigue area is known. Physical distance between two PZT sensors L_{phy} and width of the sample W_{phy} the delamination area are measured before segmentation. Finally, D_{phy} is calculated by equation 6.9

$$\frac{d_{pix}}{l_{pis} \times w_{pix}}(L_{phy} \times W_{phy})cm^2 \tag{6.9}$$

Where $L_{phy} = 10cm$ and $L_{phy} = 2.5cm$. Notice that unsupervised VMOGA generate result that contains edge effect which in practice does belong to delamination.

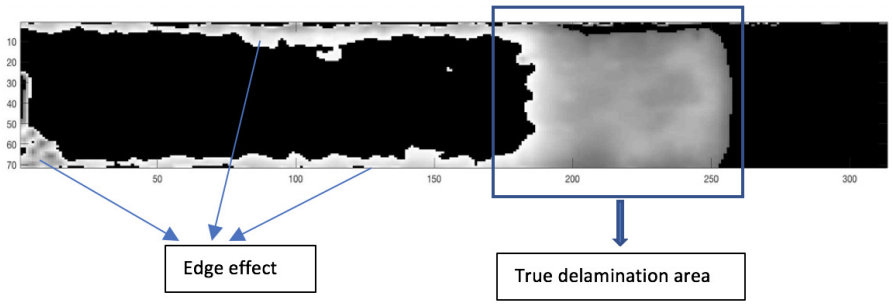

Figure 6.12 Edge effect and real delamination area in PZT cutting image of Healthy sample after 160K cycles

In order to separate edge effect from real delamination, a special window function is adopted. Recall the process in constructing refined membership function where

$$g_{k,j} = \begin{cases} 1 & if\, u_{k,j} \geq u_{l,j}, l = 1, 2, ..., K \\ 0 & otherwise \end{cases} \tag{6.10}$$

In fact, $g_{k,j}$ is the crisp version of fuzzy membership $u_{k,j}$, $W = \{w_i\}$ is predetermined as a group of rectangular windows with width equal to one. The length of w_i is denoted as l_i

$$h_{k,j} = \sum_i \sum_{j \in w_i} g_{k,j} \quad T = \sum \frac{l_i + 1}{2} \tag{6.11}$$

$$h^*_{k,j} = \begin{cases} 1 & if\, h_{k,j} \geq h_{l,j}, l = 1, 2, ..., K \\ 0 & otherwise \end{cases} \tag{6.12}$$

Largest component of ROI produced by $h^*_{k,j}$ is selected as our segmenting result. Binary images

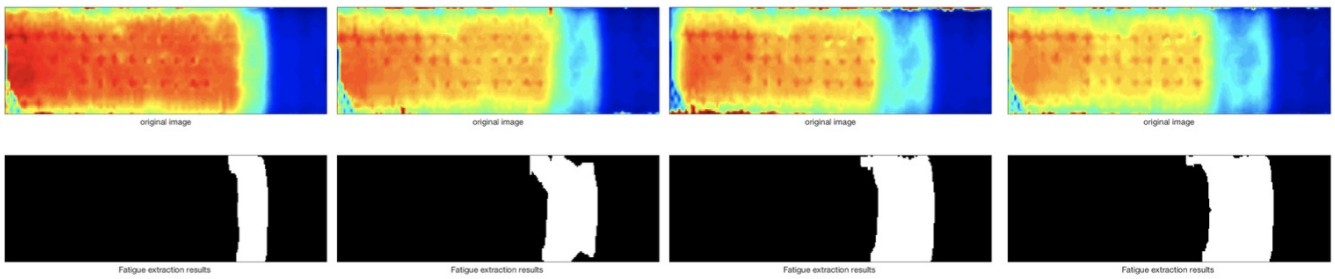

Figure 6.13 Healthy sample being subjected to Mode 1 cyclic loading after 20K, 40K, 60K and 80K cycles

denoting delamination area is shown in figure[613-614].In the end, plot of number of load cycles

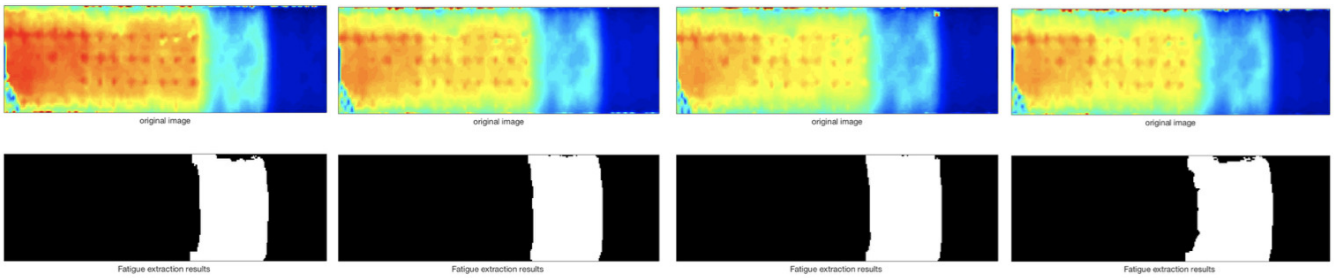

Figure 6.14 Healthy sample being subjected to Mode 1 cyclic loading after 100K, 120K, 140K and 160 cycles

versus delamination area detected is shown in fig[615]. It is clear that curve of delamination area detected by VMOGA Extraction fits very well.

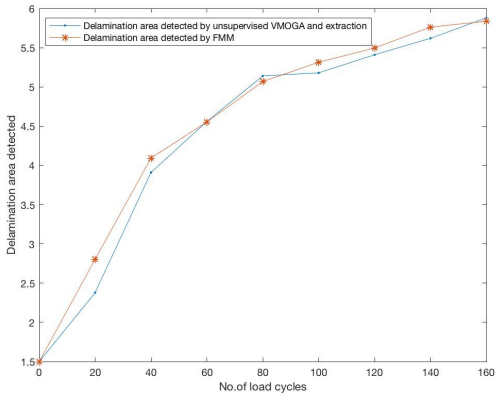

Figure 6.15 Plot of number of load cycles versus delamination area detected by VMOGA+Extraction and FMM.

CHAPTER 7

CONCLUSION AND FUTURE WORK

In this article, a variable-length multi-objective optimization technique(VMOGA) has been proposed to evolve the number of clusters automatically. NSGA-III has been adopted as underlying optimization framework for multi-objective optimization due to its capability of managing a large number of objectives. Essential components of existing multi-objective optimization algorithms have been discussed and developed. Frist of all, a new encoding and genetic operator has been designed following principle underlying binary operator in order to reduce redundant computation. Then, weaknesses with existing evolutionary operator are discussed and a rejection scheme is proposed to remove "bad" solutions that occupy population space. Furthermore, total correlation is introduced as a supplementary tool to make suitable choice of objectives. Conventional objective functions are presented. Spatial J_m, intra-cluster entropy and connectedness which incorporate various aspects of spatial information are also constructed in order to better target image segmentation problem. Superiority of VMOGA is demonstrated in segmenting MRI human brain images by comparing with several other optimization algorithms. Additionally, In the end, VMOGA is performed on detection of delamination area caused by fatigue loading in Mode I glass fiber reinforced polymer (GFRP) samples. Results suggest future potential application in fatigue detection. However, some limitations still exist in our proposed VMOGA. It is known that among the large number of cluster validity indices, none of which performs satisfactorily for a wide range of data sets alone. Therefore, it is important to select two or more objective functions that can complement and compensate for one another. This is illustrated by MRI experiment. We know that different tissues may hold similar intensity information at some locations while intensity value of a particular tissue may variate across some region. This is caused by noise and intensity inhomogeneities in data collection phase. As a result, it is possible to improve clustering results by incorporating extra spatial information. However, the amount of spatial information needed is still unknown. In future work, balance between geography-based and intensity-based objectives should be examined.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints.," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[2] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1506–1511, 2007.

[3] A. Mukhopadhyay and U. Maulik, "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with svm classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 4, pp. 1132–1138, 2009.

[4] A. Mukhopadhyay and U. Maulik, "A multiobjective approach to mr brain image segmentation," *Applied Soft Computing*, vol. 11, no. 1, pp. 872–880, 2011.

[5] J. C. Bezdek and N. R. Pal, "Some new indexes of cluster validity," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 301–315, 1998.

[6] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[7] S. Bandyopadhyay, U. Maulik, and R. Baragona, "Clustering multivariate time series by genetic multiobjective optimization," *Metron*, vol. 68, no. 2, pp. 161–183, 2010.

[8] G. N. Demir, A. S. Uyar, and S. G. Ögüdücü, "Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp. 1943–1950, ACM, 2007.

[9] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy systems*, vol. 3, no. 3, pp. 370–379, 1995.

[10] W. Wang and Y. Zhang, "On fuzzy cluster validity indices," *Fuzzy sets and systems*, vol. 158, no. 19, pp. 2095–2117, 2007.

[11] A. Ferligoj and V. Batagelj, "Direct multicriteria clustering algorithms," *Journal of Classification*, vol. 9, no. 1, pp. 43–61, 1992.

[12] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 56–76, 2007.

[13] Y. Jin, T. Okabe, and B. Sendho, "Adapting weighted aggregation for multiobjective evolution strategies," in *International Conference on Evolutionary Multi-Criterion Optimization*, pp. 96–110, Springer, 2001.

[14] D. Goldberg, "Genetic algorithm in search optimization and machine learing, addison-wesleypub," 1989.

[15] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength pareto approach," *TIK-report*, vol. 43, 1998.

[16] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[17] D. W. Corne, J. D. Knowles, and M. J. Oates, "The pareto envelope-based selection algorithm for multiobjective optimization," in *International Conference on Parallel Problem Solving from Nature*, pp. 839–848, Springer, 2000.

[18] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "Pesa-ii: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, pp. 283–290, Morgan Kaufmann Publishers Inc., 2001.

[19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[20] U. Maulik, S. Bandyopadhyay, and A. Mukhopadhyay, *Multiobjective Genetic Algorithms for Clustering: Applications in Data Mining and Bioinformatics*. Springer Science & Business Media, 2011.

[21] J.-M. Won, S. Ullah, and F. Karray, "Data clustering using multi-objective hybrid evolutionary algorithm," in *Control, Automation and Systems, 2008. ICCAS 2008. International Conference on*, pp. 2298–2303, IEEE, 2008.

[22] R.-S. Kwan, A. C. Evans, and G. B. Pike, "Mri simulation-based evaluation of image-processing and classification methods," *IEEE transactions on medical imaging*, vol. 18, no. 11, pp. 1085–1097, 1999.

[23] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 1–15, 1994.

[24] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary computation*, vol. 10, no. 3, pp. 263–282, 2002.

[25] K. Y. Yeung and W. L. Ruzzo, "An empirical study on principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[26] A. Khomenko, O. Karpenko, E. Koricho, M. Haq, G. L. Cloud, and L. Udpa, "Theory and validation of optical transmission scanning for quantitative nde of impact damage in gfrp composites," *Composites Part B: Engineering*, vol. 107, pp. 182–191, 2016.

[27] P. Banerjee, O. Karpenko, L. Udpa, M. Haq, and Y. Deng, "Prediction of impact-damage growth in gfrp plates using particle filtering algorithm," *Composite Structures*, vol. 194, pp. 527–536, 2018.