

# **Genetic Optimization for Image Segmentation**

**Maryam Gholami Doborjeh**

Submitted to the  
Institute of Graduate Studies and Research  
In partial fulfillment of the requirements for the Degree of

Master of Science  
in  
Computer Engineering

Eastern Mediterranean University  
September 2012  
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

---

Prof. Dr. Elvan Yılmaz  
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

---

Assoc. Prof. Dr. Muhammed Salamah  
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

---

Assist. Prof. Dr. Adnan Acan  
Supervisor

---

Examining Committee

1. Assist. Prof. Dr. Adnan Acan

2. Assist. Prof. Dr. Ahmet Ünveren

3. Assist. Prof. Dr. Arif Akkeleş

## ABSTRACT

The present study is concerned with optimization of image segmentation using Genetic Algorithms. The developed implementation utilizes the Split/Merge approach for image segmentation. The split portion involves K-means clustering algorithm and then a Genetic Algorithm (GA) with a proficient chromosome encoding model is applied in the merge procedure.

The goals of this study are as follows:

- a) To study different image segmentation approaches in the literature,
- b) To review the objectives of optimization in image segmentation,
- c) To conduct and implement a genetic algorithm optimization for image segmentation.

Experimental studies have shown that the above mentioned objectives are all achieved with the developed implementation.

**Keywords:** Image segmentation, genetic algorithms, genetic optimization

## ÖZ

Bu çalışma görüntü ayrıştırma işleminde genetik algoritmalar kullanılarak en iyileme yapmayı içerir. Geliştirilen uygulama parçala / birleştir yaklaşımına dayalı bir ayrıştırma yöntemidir. Parçalama işlemi için k-merkezli gruplama yöntemi kullanılırken, birleştirme yöntemi ve bununla ilgili en iyileme süreci verimli bir kromozom gösterimine sahip bir genetik algoritma ile yapılır.

Bu çalışmanın temel amaçları şu şekilde sıralanabilir:

- a) Literatürde varolan çeşitli görüntü ayrıştırma yöntemlerini tanımak
- b) Görüntü ayrıştırmada temel amaçları belirlemek
- c) Görüntü ayrıştırma işleminde en iyileme problemini bir genetik algoritma kullanarak çözmeye yönelik bir uygulama geliştirmek.

Deneysel çalışmaların sonuçları geliştirilen uygulamanın yukarıda belirtilen amaçlara ulaşıldığını göstermiştir.

**Anahtar Kelimeler:** Görüntü bölütleme, genetik algoritma, genetik eniyileme

Dedicated to  
My Beloved Family

## **ACKNOWLEDGEMENTS**

I take this opportunity to be grateful to my supervisor, Assist. Prof. Dr. Adnan Acan, for guiding me all along my thesis research and experiment. Without his priceless supervision, it would be impossible to finish this thesis.

Hereby I would like to thank my friends Reza Enayatollahi and Jamshid Tamouk, who kindly shared their knowledge and experiments with me.

Also, I appreciate the efforts of my dear friends which they have supported me in my thesis. Although, I did not mention their names; however, they will remain in my heart forever.

Last but not least, I would like to express my deepest appreciation to my family who allowed me to travel to North Cyprus and supported me during my education.

# TABLE OF CONTENTS

ABSTRACT.....	iii
ÖZ .....	iv
DEDICATION .....	v
ACKNOWLEDGEMENTS .....	vi
LIST OF TABLES .....	x
LIST OF FIGURES .....	xi
1 INTRODUCTION .....	1
1.1 Motivation.....	1
1.2 Scope and Objectives .....	2
2 LITERATURE REVIEW .....	4
2.1 Image Segmentation.....	4
2.2 Classifications .....	4
2.3 Historical Review.....	5
3 THEORY OF GENETIC OPTIMIZATION FOR IMAGE SEGMENTATION ...	11
3.1 Image Segmentation.....	11
3.1.1 Region-Based Methods .....	11
3.1.2 Edge-Based Methods .....	12
3.1.3 Region Growing Procedure.....	13
3.1.4 Split/Merge Approaches .....	14
3.1.5 Clustering Based Methods .....	14
3.2 Genetic Algorithms .....	14
3.2.1 GA's Population.....	15
3.2.2 GA Criteria.....	16

3.2.2.1 Fitness Function .....	17
3.2.2.2 Initialization of the Population.....	17
3.2.3 Selection Procedure.....	18
3.2.5 Reproduction Process.....	18
3.2.5.1 Crossover .....	18
3.2.5.1.1 One-Point Crossover .....	19
3.2.5.1.2 Two-Point Crossover .....	19
3.2.6 Termination Conditions .....	20
3.3 Application to Image Segmentation.....	20
4 THE PROPOSED APPROACH AND ITS IMPLEMENTATION.....	22
4.1 Algorithm Description .....	22
4.2 Split Process.....	25
4.2.1 K-means Clustering Algorithm.....	25
4.2.2 Connected Component Labeling.....	28
4.3.2.3 Region Adjacency Graph.....	29
4.3 Merge Procedure Using the GAs .....	30
4.3.1 Chromosome Encoding Method .....	31
4.3.2 Population Generation.....	32
4.3.2.1 Neighbor Selection for Merge Procedure .....	32
4.3.2.2 Dissimilarity Distance Computation.....	33
4.3.3 Fitness Function .....	35
4.3.4 Selection Operator in GAs .....	36
4.3.4.1 Roulette Wheel Selection.....	37
4.3.5 Crossover Operation .....	39
4.3.6 Mutation Operation .....	43



4.3.7 Elitism Strategy .....	43
5 RESULTS AND DISCUSSION .....	51
5.1 Evaluation of GOIS Approach on Noisy Images.....	52
5.1.1 Split Procedure Results .....	53
5.1.2 Merge Procedure Results Based on GA Optimization.....	55
5.2 Evaluation of GOIS Approach on Images with Different Characteristics.....	59
5.2.1 Split Procedure Results .....	60
5.3 Comparison of GOIS Approach with Other Approaches .....	64
5.4 Population Size .....	65
5.5 Stability of the Algorithm .....	70
6 CONCLUSIONS AND FUTURE WORK .....	71
6.1 Conclusions.....	71
6.2 Future Work.....	72
REFERENCES .....	73

## LIST OF TABLES

Table 5-1. Split Results of Three Different Noises on Lena Image.....	55
Table 5-2. Comparison of CPU Times for the Three Different Lena Images Resulting of GOIS .....	59
Table 5-3. Comparison of Three Images Resulting from the Split Procedure.....	61
Table 5-4. Comparison of Three Different Images Resulting of Merge Procedure ...	63
Table 5-5. Comparison of the Performance Resulting from Different Images with Different Population Size.....	66
Table 5-6. Computing the Total Performance of Population Size= 25.....	67
Table 5-7. Comparison of FCM, SGA and GOIS Approaches Using Various Images .....	68

# LIST OF FIGURES

Figure 2-1.Classification of Image Segmentation Techniques .....	5
Figure 2-2.The Original Gray Image as an Input to Algorithms in [3].....	7
Figure 2-3.Segmentation Results of Algorithms Proposed in [3].....	8
Figure 2-4.Comparison of Three Different Images: (a)-(c) Source Images as Inputs to Algorithm in [22]; (d)-(e) Segmented Results of Algorithm Proposed in [22] .....	9
Figure 2-5.(a) Original Image of Heart as Input to Approaches in [23]; (b) Initial Contour of Heart Image .....	10
Figure 3-1.Region Growing Iterations of the Brain Image in [28] .....	13
Figure 3-2.The GA Flowchart.....	16
Figure 4-1.(a) Original Input Image; (b) Gray-Scale Image; (c) Preprocessed Image .....	24
Figure 4-2.Preprocessing and Split Procedure Flowchart.....	25
Figure 4-3.(a) Original Gray-Level Image; (b) Histogram of the Original Image ....	27
Figure 4-4.Clustered Image Resulting from K-means Algorithm, (b) Histogram of the Clustered Image .....	28
Figure 4-5.(a) Segmented Image into Separated Regions; (b) RAG of Segmented Image.....	29
Figure 4-6.Optimization Image Segmentation Process .....	30
Figure 4-7.Chromosome Encoding Model of Separated Regions .....	31
Figure 4-8.Initial Individual Structure .....	32
Figure 4-9.A Roulette Wheel Marked for Five Individuals According to Their Fitness Values .....	38
Figure 4-10.Two Selected Chromosome as the Parents .....	40

Figure 4-11 An Example of the Crossover Segment on Parents.....	41
Figure 4-12.Swapped Crossover Segment between Two Parents. ....	42
Figure 4-13: Representation of the Crossover Operator .....	42
Figure 4-14.Results of the Mutation Operation .....	43
Figure 4-15.An Example of Chromosome Representation of Segmented Image.....	44
Figure 4-16.The Subsets of RAG.....	45
Figure 4-17.Merge Procedure in Segmented Image: $r_3$ and $r_4$ are Merged .....	46
Figure 4-18.Subsets of RAG: (a): Neighborhood of $r_5$ before Merge Procedure; (b): Neighborhood of $r_3$ after Merge Procedure by Updating the RAG.....	47
Figure 4-19.Merge Procedure in Segmented Image: $r_5$ and $r_6$ are Merged .....	47
Figure 4-20: Subsets of RAG: (a): Neighborhood of $r_8$ before Merge Procedure; (b): Neighborhood of $r_8$ after Merge Procedure by Updating the RAG.....	48
Figure 4-21.Merge Procedure in Segmented Image .....	48
Figure 4-22.New Chromosome Representation after Merge Procedure.....	49
Figure 4-23.An Example of Initial Population Includes 10 Different Individuals and Their Computed Fitness Values.....	50
Figure 5-1.Test Image: (a) Original Gray Lena Image; (b) Clustered Image Using K- means Algorithm; (c) Separating Image by Drawing Boundary; (D) Region Detection.....	53
Figure 5-2.Test Image: (a) Corrupted Image with Gaussian Noise(Variance= 0.001 and Mean= 0.0); (B) Clustered Image Using K-Means Algorithm; (C) Separating Image by Drawing Boundary; (D) Region Detection .....	54
Figure 5-3.Test Image: (A) Corrupted Image with Gaussian Noise(Variance= 0.05 and Mean= 0.0); (B) Clustered Image Using K-Means Algorithm; (C) Separating Image by Drawing Boundary; (D) Region Detection .....	54

Figure 5-4.Merge Procedure of Original Lena Image after Various Numbers of Generations .....	56
Figure 5-5.Merge Procedure of Corrupted Lena Image by Gaussian Noise (variance=0.001 and mean=0) after Various Number of Generations.....	57
Figure 5-6.Procedure of Corrupted Lena Image by Gaussian Noise (variance= 0.05 and mean=0) after Various Number of Generations.....	57
Figure 5-7.Comparison of the Fitness Values through the Various Generations .....	58
Figure 5-8.Test Image: (a) Original Gray-Level Image; (b) Clustered Image Using K-Means Algorithm; (c) Separating Image by Drawing Boundary; (d) Region Detection .....	60
Figure 5-9.(a) Original Gray Fruit Image; (b) Clustered Image Using K-Means Algorithm; (c) Separating Image by Drawing Boundary; (d) Region Detection.....	61
Figure 5-10.Merge Procedure of Original Fruit Image after Various Numbers of Generations .....	62
Figure 5-11.Merge Procedure of Original Yacht Image after Various Numbers of Generations .....	62
Figure 5-12.Comparisons of Four Image Segmentation Approaches: (a)-(c) Source Images with the Size of 256×256; (d)-(f) Results of Canny Segmentation; (g)-(i) Results of FCM in [1]; (j)-(l) Results of SGA ; (m)-(o) Results of GOIS.....	64
Figure 5-13.Stability of Various Fitness Values.....	70

# Chapter 1

## INTRODUCTION

### 1.1 Motivation

Image segmentation is one of the most significant tasks in image processing [1]. So far, a massive number of segmentation algorithms with different segmentation criteria have been implemented. The main goal of image segmentation is to detect any objects of an image and distinguish them from the background [2]. Segmentation can be used for object recognition, boundary detection within stereo or motion systems, image compression, image editing, or image retrieval from databases [3]. Worldwide demand for accurate object detection reveals a pressing need to develop optimization techniques for image segmentation. These techniques are basically used in different applications such as medical images to correctly detect abnormalities and for face detection in criminal purposes. A segmentation algorithm splits an image into separated regions, where each region is a set of connected similar pixels. In order to detect a region, the entire pixels of the region must be comparable in some features such as gray-level intensity value, texture, luminance, chrominance and other specific characteristics that can specify homogeneity of the region. In addition to accuracy of image segmentation, time consumption has high level of importance. Different results can be obtained from different techniques of image segmentation according to the relevant cost function or quality measure of the segmented image. As a result, optimization techniques can be utilized to generate an optimum outcome

in comparison with simple segmentation methods. There are several optimization techniques on image segmentation; all with the goal of defining the best solution [4].

## **1.2 Scope and Objectives**

The main intention of the current work is to segment an image and optimize the result according to some predefined quality measures. The input for a segmentation algorithm is an image that is converted to a gray-level intensity image through the preprocessing techniques. According to our goal and the expected results, preprocessing phase can have different steps; the raw intensity image should be transformed to gray-level intensity image and then pass through noise reduction, edge enhancement and contrast enhancement are also main preprocessing steps that can be applied on the input image. After preprocessing steps, a segmentation algorithm is applied with the aim of splitting the preprocessed image into several homogeneous regions. For the purpose of optimized segmentation, we should define a cost function or quality measure for our approach. Consequently, a significant plan to gain optimum results is to minimize the value of the cost function. In this study, in order to optimize image segmentation, an evolutionary computation algorithm is utilized.

The remaining part of this thesis is organized as follows:

In Chapter 2, the previous experiments and researches on the image segmentation techniques, which are most relevant to the presented study, are reviewed and explained very briefly.

The principals for each step of the evolutionary algorithm for image segmentation procedure are discussed theoretically in Chapter 3.

Chapter 4 discusses the algorithms and the procedures of Genetic Optimization for Image Segmentation (GOIS) technique under the presented study.

Besides, chapter 5 presents the results of the presented work, in detail.

More to the point, Chapter 6 introduces the comparisons on the results of this study and the suggestions for future work are presented.



## **Chapter 2**

### **LITERATURE REVIEW**

#### **2.1 Image Segmentation**

Image segmentation is an essential procedure for object detection in any image, video or computer vision applications. Comprehensive researches have been performed on generating many different algorithms for image segmentation. However, it is still difficult to judge which algorithm results in a better segmentation outcome compared to the others. Image segmentation approaches are used for partitioning an image into sets of homogeneous pixels [1] where each set is known as a region corresponding to a real object. It is fruitful to mention that in a segmented image, the adjacent regions cannot be homogeneous [2]. Furthermore, there should not be any intersection between segmented regions of the image. Objects of an image can be distinguished from their background by using a successful segmentation algorithm. In each region, pixels are similar to each other with respect to some feature such as intensity, color or texture of the image [3].

#### **2.2 Classifications**

Typically, image segmentation techniques can be categorized into four main groups: clustering-based methods, edge-based approaches, region-based approaches and split/ merge approaches [1]. A graphical illustration of various segmentation approaches is given in Figure 2-1.

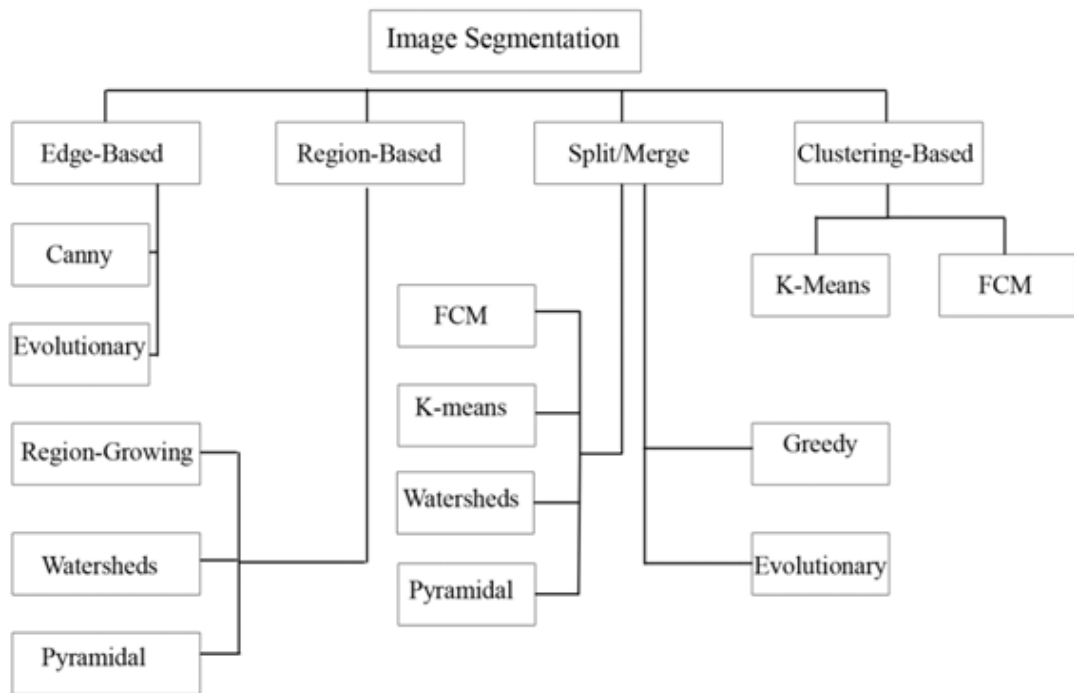


Figure 2-1. Classification of Image Segmentation Techniques [1]

Edge-based image segmentation can be done by means of two main techniques; canny edge detection which is based on a predefined threshold value and evolutionary edge detection methods [5]. Region-based approaches can be categorized to Region Growing [6], [7], [8]; Watersheds [9], [9], [11]; and Pyramidal [12]. For split/merge group of algorithms, there are four split approaches; Pyramidal Segmentation [12]; Watersheds [9], [9], [11]; FCM [13] and K-means [13]. In merge process, Region Adjacency Graph (RAG) [14] and Nearest Neighbor Graph [9] representations are two widely used and offered structures. Moreover, there are two clustering-based approaches, namely: the Fuzzy-C-Means [13], [16], [17] and K-Means clustering algorithms [13].

### 2.3 Historical Review

In the last 3 decades, in order to detect distinct objects in an image, a number of helpful algorithms have been implemented by the computer vision association. A

variety of experiments, researches and papers have been developed on image segmentation process from different aspects such as active contour worked out by Isard and Blake et al in “1998”, snake methods introduced by Kass et al in “1988” and shortest path methods carried out by Mortensen and Barrett in “1998”. It is obvious that different approaches were defined with various cost functions to attain a desired segmentation. In the following paragraphs, some of the most relevant studies to this thesis will be mentioned and explained briefly.

In “1999” Suchendra M. Bhandarkar and Hui Zhang [3], studied the image segmentation using metaheuristics. Three categories of hybrid stochastic optimization approaches were used to minimize an objective function. The first component of their hybrid approaches is one of a simulated annealing (SA) [18]; random cost algorithm (RCA) [19], [20]; and micro-canonical annealing (MCA) [21], which are the subcategory of stochastic hill-climbing search techniques, and they can be characterized by their ability to get away from local optima in the solution space. The second component is taken as the genetic algorithms. Their search procedures employ a combination of the building blocks property of the genetic algorithms and hill-climbing techniques to take advantages from both methods. Their paper examined three hybrid evolutionary algorithms in image segmentation:

- SA–GA: a combination of the GA and SA originally proposed by Mahfoud and Goldberg [22];
- MCA–GA: a combination of the GA and MCA proposed in their paper;
- RCA–GA: a combination of the GA and RCA proposed in their paper.

Their experimental results exhibit high visual quality of the final segmented image due to global optimization based segmentation and low implementation time as a result of the hybrid evolutionary algorithms. A Comparison of different segmentation methods proposed by the authors is illustrated in Figure 2-2 and 2-3.



Figure 2-2. The Original Gray Image as an Input to Algorithms in [3]

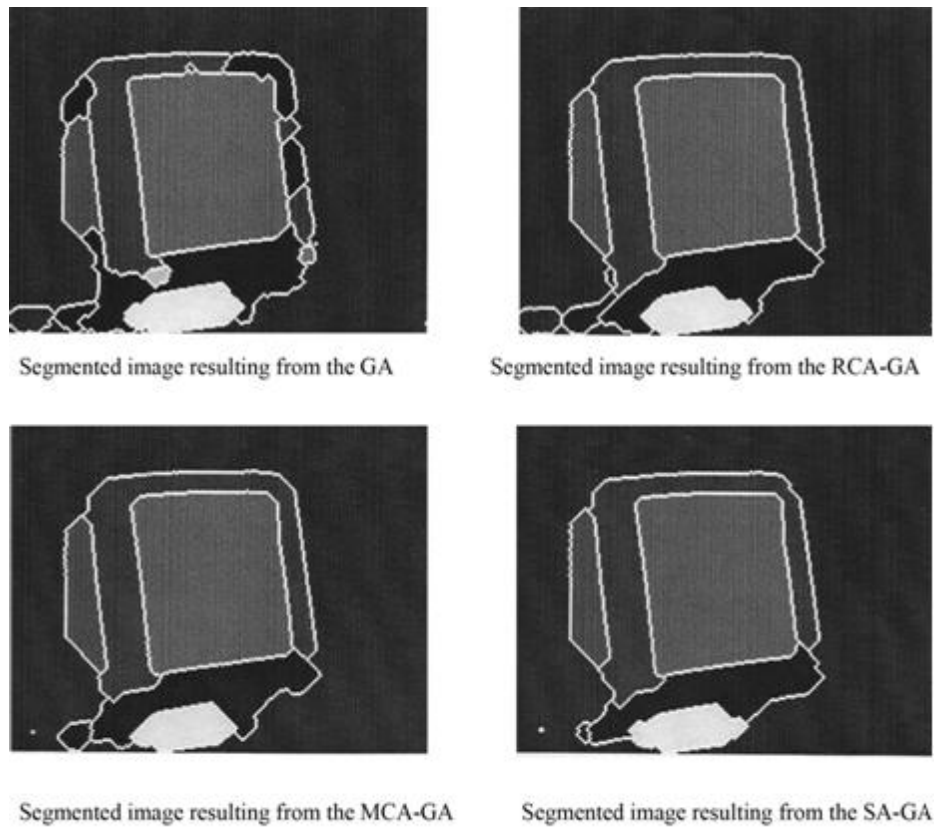


Figure 2-3. Segmentation Results of Algorithms Proposed in [3]

In “2008”, Mohamed Ben Salah, Ismail Ben Ayed and Amar Mitiche have reviewed different methods of optimization for image segmentation [23]. They represented a segmented image by a graph where each node represents a collection of connected pixels in a region. This graph also involves two additional nodes called terminals. Graph cut optimization methods were used to optimize the image segmentation problem. A solution of the objective function minimization is represented by minimization of an energy function. The minimum cut problem consists in finding the cut ( $C$ ) in a given graph with the lowest cost. The cost of a cut,  $|C|$ , is the sum of its edge weights from start point to terminal point. This min-cut optimization approach results in the best segmentation due to cost function minimization. The region merging process used in this approach allows segmentation of the images into

the most appropriate number of regions and one gets closely the same energy and region parameter at convergence in reasonable running time. Their experimental results are illustrated in Figure 2-4.

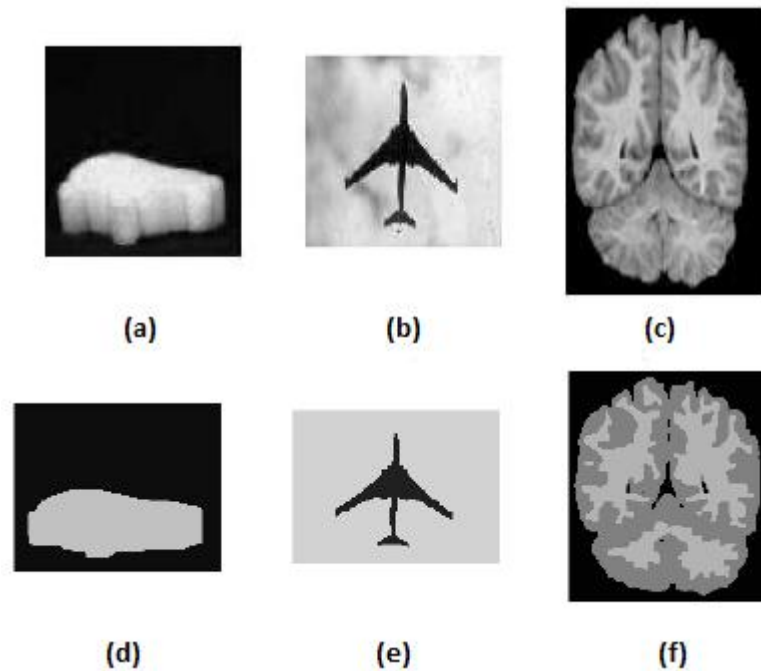


Figure 2-4. Comparison of Three Different Images: (a)-(c) Source Images as Inputs to Algorithm in [22]; (d)-(e) Segmented Results of Algorithm Proposed in [22]

In 2010, an essential search algorithm on Ant Colony Optimization (ACO) for image segmentation was done by Y. Feng and Z. Wang [24]. ACO has been applied effectively to several combinatorial optimization problems such as Travelling Salesman Problem (TSP) by Dorigo and Gambardella in 1997, Vehicle Routing Problem by Bullnheimer et al. in 1998 and Flow Shop Problem worked out by Stutzle in 1998 and etc. The principal behind ant colony optimization algorithm is the real ant behavior. Ants are walking randomly in order to find food. Once food source is found, ants trace pheromone hormone on the ground as guidance to the food location. When the ants sense pheromone trail on the ground they could follow

the trail and enhance the shortest route by tracing pheromone more frequently. Several authors studied the edge detection problem and proposed solutions by ant colony search algorithm. The experimental results of their segmentation approach are shown in Figure 2-5.

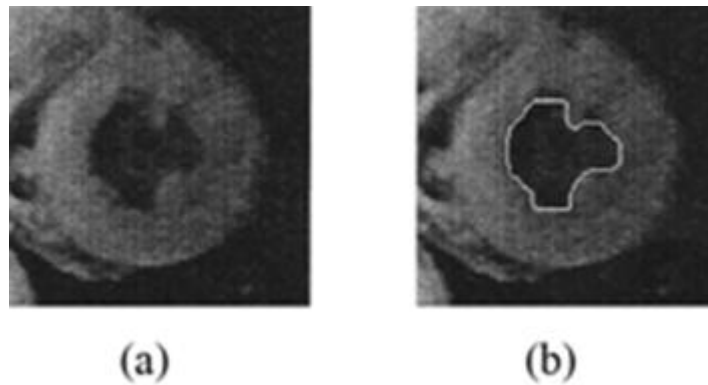


Figure 2-5.(a) Original Image of Heart as Input to Approaches in [23]; (b) Initial Contour of Heart Image

## Chapter 3

# THEORY OF GENETIC OPTIMIZATION FOR IMAGE SEGMENTATION

### 3.1 Image Segmentation

Image segmentation has long been known as a difficult problem in image processing and many different approaches have been proposed to produce the desired results. In computer vision, Segmentation is the method of dividing an image into several meaningful regions for applications such as objects tracking and understanding [25]. The aim of segmentation is to simplify and modify the demonstration of an image into something that is more significant and easier to investigate [1]. Image segmentation is normally used to detect different objects and their boundaries in images. In particular, image segmentation is the procedure of conveying a label to each pixel such that pixels with similar labels share firm visual characteristics. As it is mentioned previously, there are several segmentation methods, such as Edge-based, Region-based, Clustering-based methods and Split/merge approaches [1]. Brief descriptions of these approaches are given in subsequent sections.

#### 3.1.1 Region-Based Methods

In region-based methods, the input image is divided into several connected regions by grouping the adjacent pixels of close intensity levels [1]. The goal is the detection of regions that satisfy certain predefined homogeneity criteria. Neighboring regions are further merged due to their homogeneity or sharpness of region boundaries.



Region-based segmentation is said to be a technique for determining the region directly. The two basic constraints to Region-Based Segmentation are as follows:

$$\bigcup_{i=1}^n R_i = R \quad (3-1)$$

where  $R_i$  is a connected region,  $i = 1, 2, \dots, n$ . That means union of all regions must be the original image  $R$  [1].

$$R_i \cap R_j = \emptyset, \text{ for } i = 1, 2, \dots, n \quad (3-2)$$

The second constraint means that every pixel should belong to only one specific region.

### 3.1.2 Edge-Based Methods

Edge-based methods of image segmentation are standing on edge detection which is a well-expanded field in image processing. Since there is often a sharp alteration at intensity of the object boundaries, these boundaries can be considered as the edges. However, closed region boundaries must be determined to detect image objects and the detected edges are the boundaries between objects [26]. Therefore, edge detection techniques have been used as the basic step of other segmentation methods. In this approach, image edges are recognized and then linked into lines that indicate the borders of image objects [26]. Candidate edges are extracted by thresholding the gradient or Laplacian magnitude [27]. It is probable that the edges that are recognized by edge detection algorithms are sometimes discontinuous. In order to solve this problem several evolutionary algorithms have been proposed for detecting well-localized and continuous thin edges based on optimization of edge configurations [5].

### 3.1.3 Region Growing Procedure

Region growing algorithms have been established as effective approaches for image segmentation. The principle of a region growing algorithm is to start from a seed of the global region (typically one or more pixels), which is inside the object to be segmented. The pixels that surrounded this region are also evaluated to decide whether they should be considered as a part of this region or not. If so, they are added to the region until the last such pixels is evaluated [6]. In a growing region, a simple criterion for including pixels is to evaluate intensity value inside a specific interval [8]. Most of the algorithmic complexities of region growing methods come from visiting the neighboring pixels. For instance, in Figure 3-1, region growing has been started from a seed in (a) and it could detect all pixels of the brain image after various numbers of iterations in (f) [28].

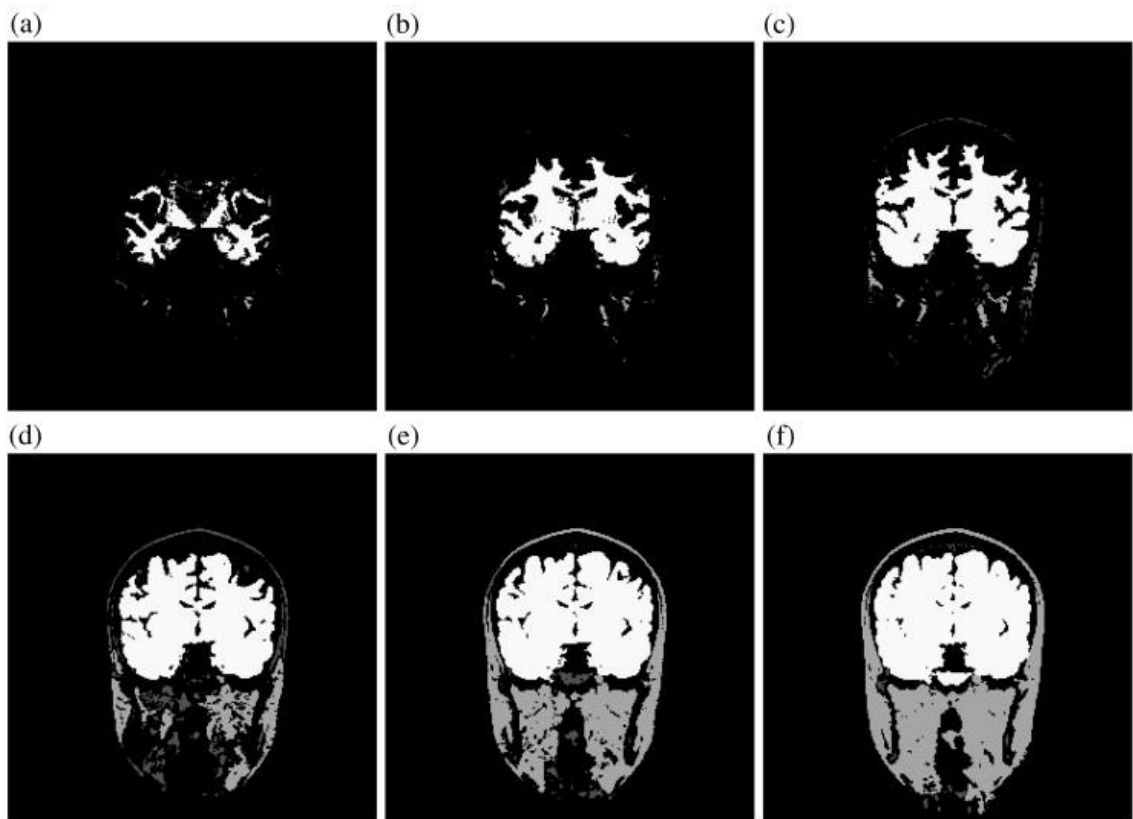


Figure 3-1. Region Growing Iterations of the Brain Image in [28]

### **3.1.4 Split/Merge Approaches**

As the name implies, these segmentation approaches have two separated procedures; segmenting the image into several regions and then merging some inappropriate regions into the correct ones. In split/ merge methods, an input image is segmented into set of homogeneous regions in advance [1]. Then, similar neighboring regions are merged according to certain homogeneity features. In merge phase, each inappropriate regions need to be merged with one of their neighbors. As a result, Region Adjacency Graph (RAG) is proposed to determine neighboring regions [3]. After each merge procedure's iteration, RAG must be updated to represent the new regions and their neighbors correctly. The concept of RAG will be explained in details in 4.3.2.3.

### **3.1.5 Clustering Based Methods**

In these methods, the entire pixels of an image are sorted in a histogram according to their intensity values [1]. Then, a predefined number of clusters will be defined to split the intensity histogram into several intervals [29]. In this unsupervised region segmentation algorithm, pixels which are located at the same cluster may not be adjacent. Therefore, the number of regions is not the same as the number of clusters [1]. There are several clustering based methods such as K-means clustering and fuzzy-c-means (FCM). In this study, K-means algorithm is used and it is explained in details in section 4.2.1.

## **3.2 Genetic Algorithms**

Genetic Algorithms (GA) are stochastic search procedures introduced by J.Holland in the 70's [30]. These algorithms are based on ideas and techniques from genetic and evolutionary theory which is a field of artificial intelligence and is a kind of searching for good solutions that mimics the process of natural evolution [31], [30],

[32]. GAs generate valuable solutions for hard optimization problems using techniques that are inspired by natural evolutionary operators, such as inheritance, mutation, selection, and crossover [3].

### **3.2.1 GA's Population**

In GAs, there is a population containing a number of solutions which are represented by strings (called chromosomes or the genotype) that evolve in the direction of better solutions [3]. Each string is an encoded candidate solution. Conventionally, solutions are encoded in binary strings of 0s and 1s, but other kinds of encoding models are also probable. The evolution starts by generating several individuals to create an initial population. Then, the fitness function is computed for each individual to produce a selection priority for individuals throughout the generations. Therefore, individuals are preferred from the present population according to their fitness values and modified to a number of offspring. The new population replaces the current population and is used as an input to the next iteration of the algorithm. Usually, the algorithm will be terminated when either maximum number of generations is reached, or a reasonable fitness value is attained.

Figure 3-2 illustrates GA flowchart through the various numbers of generations.

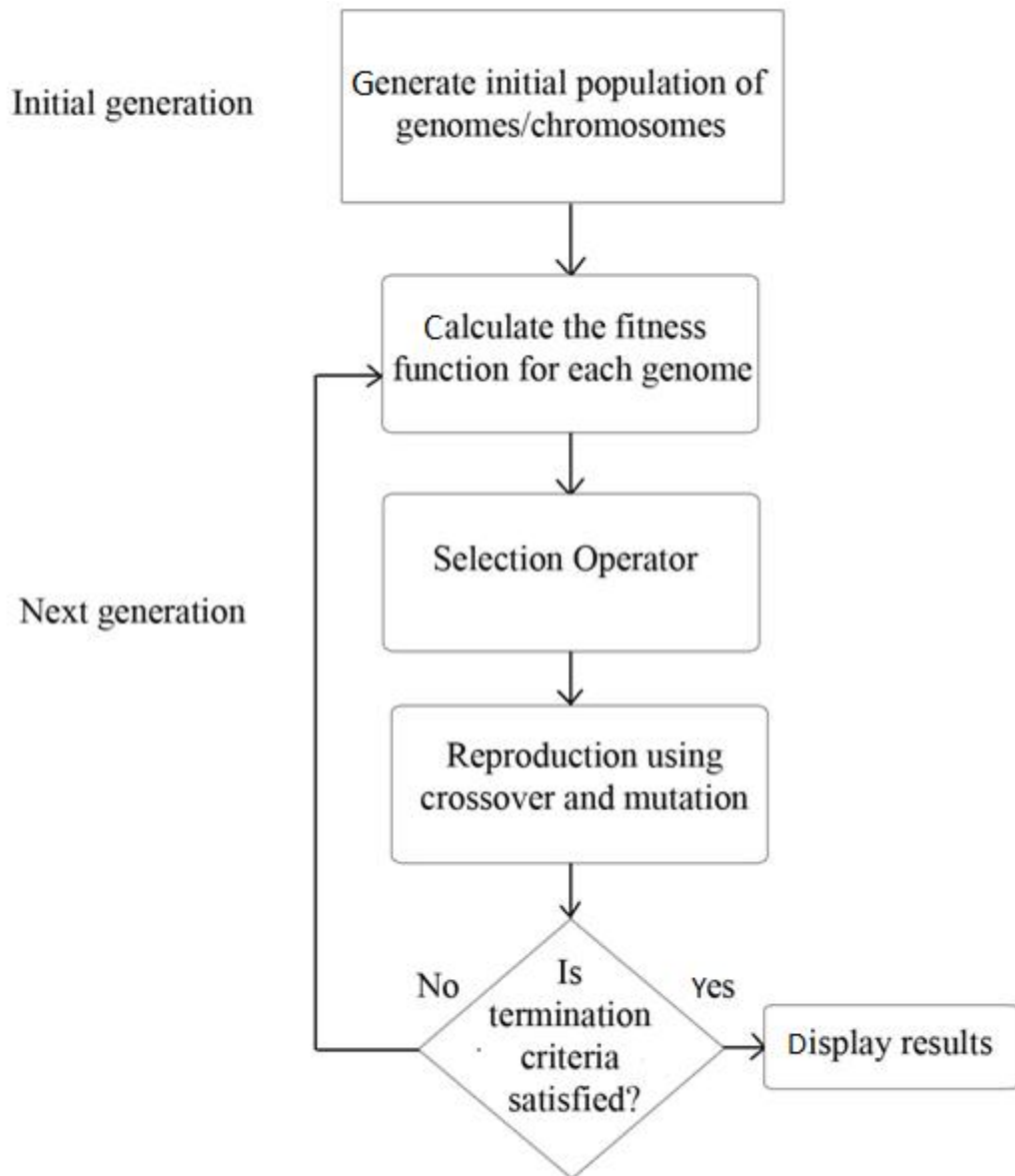


Figure 3-2. The GA Flowchart

### 3.2.2 GA Criteria

A common genetic algorithm involves two main parts:

1. All solutions should have a genetic representation (in a shape of chromosome)

2. There should be a fitness function to assess the solutions,

An array of binary bits is a typical representation of a solution. However, other types of representation can also be used [31].

### **3.2.2.1 Fitness Function**

Regarding optimization problems, to produce better solutions from the current one, there should be a fitness function to evaluate the quality of each encoded solution through the generations. The form of the fitness function is always a problem-dependent issue [33]. Fitness value is a quality measurement of each solution. Better fitness values belong to better individuals in each population. When termination criteria are satisfied, algorithm reaches to better fitness value. In the final generation, a solution with better fitness value among others is found as the desired solution. As an example of computing the fitness value, knapsack problems [34] are explained here. The best solution for these problems is a representation that maximizes the total value of the objects that can be located in a knapsack with a limited capacity. This solution might be illustrated by a binary array of bits. Each object is represented by one bit, which has the value of 0 or 1. Value of 0 means object is not selected and 1 means object is selected to put in the knapsack. Putting objects in knapsack continues until the capacity of knapsack becomes full or it would not be enough for next object. As a result, several random individuals are generated. The fitness of each solution is the sum of values of objects that have a value of 1 in their related bit of the binary array or the chromosome.

### **3.2.2.2 Initialization of the Population**

For the purpose of creating an initial population, many individuals (solutions) are generated randomly. Usually, the population size involves hundreds or thousands of probable solutions. Subsequent to generate first population and define fitness

function, a new population is produced through the mutation, crossover, inversion and selection operators.

### **3.2.3 Selection Procedure**

An important procedure during GA generations is how to select individuals from the current population and use them to generate next population. Selection operation directly depends on the individual quality. The quality of each individual is measured by a fitness function. Individuals with better fitness value are more likely to be selected [3]. There are several selection algorithms that can prefer better individuals from a population such as roulette wheel selection and tournament selection operators.

### **3.2.5 Reproduction Process**

Reproduction methods are biologically inspired and based on the use of two parents to produce a child. To produce each new child, two individuals (parents) are selected from the current generation to reproduce new individuals for next generation through the crossover operation and/or mutation. Inherently, a generated child shares many characteristics of its parent through the swapping of some genes between parents. Therefore, the new child takes advantages from both parents and might be better in comparison to the previous individuals. Reproduction process will be terminated when the size of the new population is equal to the size of the last population.

#### **3.2.5.1 Crossover**

Crossover is a significant way to mate two parents for the production of a new child (offspring). Two individuals are selected through the selection operation and then used as two parents [34]. In crossover operation, random numbers of genes are swapped between parents. The main advantage of this operation is that the generated child is better than its parents if it takes the best properties from its parents. The most

popular kinds of crossover operation are the one point crossover, two point crossover, uniform crossover and intelligent crossover operation. Two types of crossover operators are described briefly as follows:

#### **3.2.5.1.1 One-Point Crossover**

In this method a crossover point is selected randomly within a chromosome of parents, and then the bits of two parents are swapped from this point [35]. Consequently, two new offspring are produced. In following, two parents are chosen for one point crossover operation. The “|” symbol indicates the randomly selected crossover point.

Parent 1: 11001|010

Parent 2: 00100|111

After interchanging the bits between parent chromosomes at the crossover point, the following offspring are produced:

Offspring1: 11001|111

Offspring2: 00100|010

#### **3.2.5.1.2 Two-Point Crossover**

In this method two crossover points are selected randomly within each parent chromosomes [35]. Genes in the interval of two points are swapped between two parents.



In the following example, two parents are chosen for two point crossover operation:

Parent 1: 11|001|010

Parent 2: 00|100|111

After swapping the bits between parent chromosomes in the interval of two crossover points, the following offspring are produced:

Offspring 1: 11|100|010

Offspring 2: 00|001|111

### 3.2.6 Termination Conditions

GAs have a repetitive procedure until the termination condition is reached. Some of the necessary termination conditions are mentioned as follows [31]:

- Reach to a solution which has the predefined reasonable fitness value.
- Reach to the maximum number of generations.
- Reach to a predefined allocated budget (computation time/money).
- Combinations of the above.

### 3.3 Application to Image Segmentation

Recently, a variety of evolutionary optimization algorithms has been introduced in order to optimize the image segmentation results. In order to achieve this aim, genetic algorithms which are vital components of the wider class of evolutionary algorithms are utilized [31]. Therefore, an image segmentation method with high accuracy and localization ability is generated. In this method, the initial segmented

image is represented by means of a chromosome, where each gene of the chromosome indicates a separated region of the image. In order to generate the primary population, several chromosomes will be produced based on the initial chromosome. The other populations will be generated accordingly.

## Chapter 4

# THE PROPOSED APPROACH AND ITS IMPLEMENTATION

### 4.1 Algorithm Description

The presented evolutionary image segmentation approach consists of three phases: preprocessing filters split procedure and merge procedure using genetic optimization. In the first step of our segmentation approach, original input image should be transformed into a gray-level intensity image.

Preprocessing phase includes the following steps:

1. To convert the original input image into a gray-level intensity image using the Matlab code as:

```
Gray_Image= rgb2gray(Original_Image);
```

Conversion of a color image to grayscale is not unique. A common strategy is to match the luminance of the grayscale image to the luminance of the color image. To convert any color to a grayscale representation of its luminance, first one must obtain the values of its red, green, and blue (RGB) primaries in linear intensity encoding. When converting an RGB color image to grayscale all the color is replaced with shades of gray. Therefore, image is no longer in RGB.

2. To reduce the level of noise in the gray-level intensity image using the

Matlab code as:

```
Gray_Image=medfilt2 (Gray_Image, [3 3]);  
G=fspecial('gaussian',[5 5],2);  
Gray_Image = imfilter(Gray_Image,G,'symmetric','conv');
```

It is a process that attempts to restore or recover or renovate an image that has been degraded or corrupted by some degradation events using above noise filters [36].

3. To enhance the dissimilarity of the edge pixels using the Matlab code as:

```
h=fspecial('unsharp');  
Enhanced_Image=imfilter(Gray_Image,h);  
Enhanced_Image =imfilter(Gray_Image,h,'replicate');  
Imshow(Enhanced_Image);
```

Edge enhancement is an image processing filter that enhances the edge contrast of an image in an attempt to improve its acutance (apparent sharpness). The filter works by identifying sharp edge boundaries in the image, such as the edge between a subject and a background of a contrasting color, and increasing the image contrast in the area immediately around the edge.

Figure 4-1, indicates the differences between the original input image, gray-scale image and the preprocessed image.



Figure 4-1.(a) Original Input Image; (b) Gray-Scale Image; (c) Preprocessed Image

The split procedure includes the following steps:

1. To cluster pixels of the image by K-means clustering algorithm as explained in section 4.2.1.
2. To identify different regions using connected component labeling as explained in section 4.2.2.
3. To determine region neighbors by means of the Region Adjacency Graph (RAG) as explained in section 4.3.2.3.
4. Removal of small regions in order to prevent over-segmentation.

These steps are briefly explained and exemplified in the following sections.

Preprocessing and split procedure are illustrated in Figure 4-2.

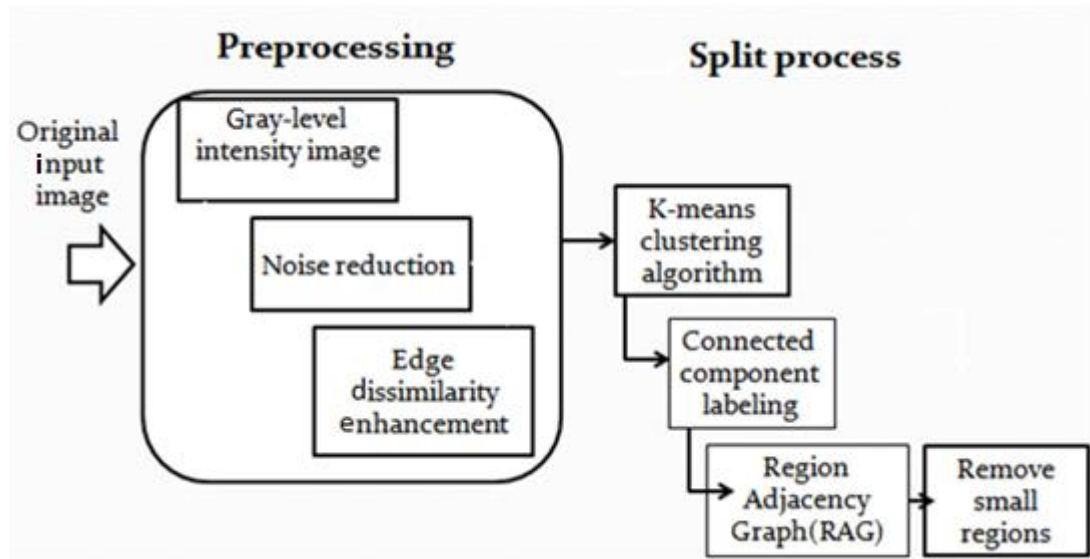


Figure 4-2.Preprocessing and Split Procedure Flowchart

After applying the preprocessing steps and the split procedure, an initial segmented image is formed which is subjected to optimized merge procedure by means of a genetic algorithm.

## 4.2 Split Process

In this study, in order to cluster the entire pixels of the image into several groups, the original input image should be divided into several regions using the K-means clustering algorithm.

### 4.2.1 K-means Clustering Algorithm

K-means is an applicable method to divide the image pixels into a variety of clusters based on histogram technique which is a representation of number of pixels in each intensity level [36]. K-means algorithm categorizes pixels together whose features are similar to each other. In the first iteration of K-means algorithm, a number of predefined pixels are selected randomly as the center for each cluster. The final result

illustrates several clusters that every cluster indicates a specific interval in the histogram.

In our experiment, the split process consists of the following steps based on the K-means clustering algorithm.

Step 1: choose  $K$  numbers of initial values  $V_k$  for an image with 256 gray levels using Equation 4-1.

$$V_k = \frac{256}{k+1}, \quad k = 1, \dots, K. \quad (4-1)$$

Step 2: choose pixel  $x$  as a member of cluster  $j$  if the distance between this pixel and the center of cluster  $j$  is smaller than the distance between other clusters' center, which is denoted by Equation 4-2.

$$x \in C_j \text{ if } |x_g - V_j| \leq |x_g - V_i| \quad (4-2)$$

*for*  $i = 1, 2, \dots, k, \text{ and } i \neq j.$

where  $x_g$  is the intensity value of pixel  $x$ ,

Step 3: recalculate the cluster center  $V_k$  for all clusters  $C_k$  using the Equation 4-3.

$$\text{New cluster center}(V_k) = \frac{1}{|C_k|} \sum_{s \in C_k} X_s, \quad k = 1, \dots, K. \quad (4-3)$$

where  $s$  is a member of  $C_k$ ,  $X_s$  is the gray value of pixel  $s$  and  $|C_k|$  is the number of members in  $C_k$ .

Step 4: Termination condition is reached when the entire cluster centers  $V_k$  remain constant. Otherwise, go back to the step 2.

In Figure 4-3, histogram of the original image is shown, which defines the number of pixels in each intensity level. Figure 4-4 represents the number of pixels in each cluster interval.

After applying K-means algorithm on the image, its histogram will change according to the number of clusters. By defining 4 centers in K-means approach, all pixels of the image are divided into 4 clusters utilizing the same value corresponding to the mean value of cluster. As a result, intensity distribution is only among these 4 clusters.

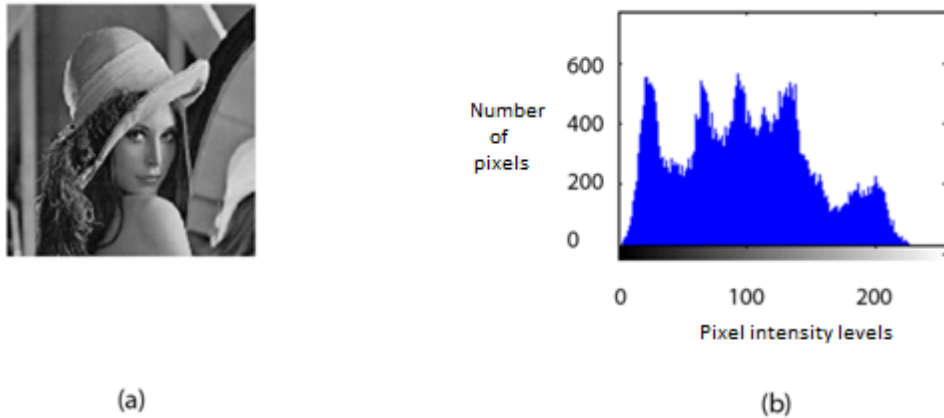


Figure 4-3.(a) Original Gray-Level Image; (b) Histogram of the Original Image



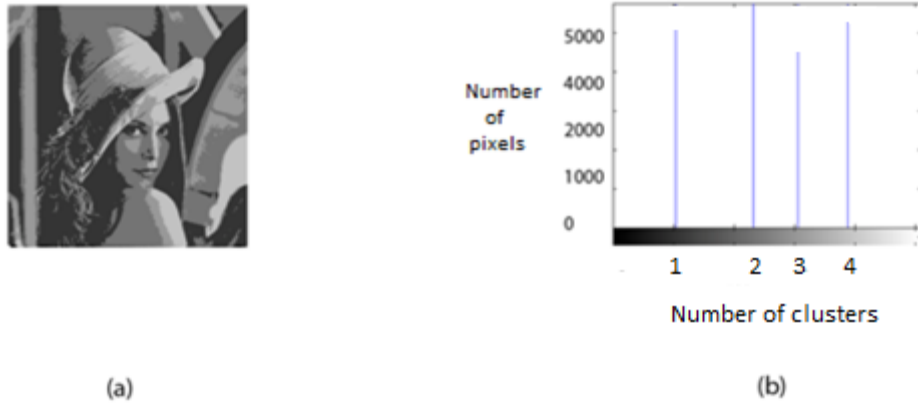


Figure 4-4. Clustering Image Resulting from K-means Algorithm, (b) Histogram of the Clustered Image

The main benefit of this technique is to avoid the complex threshold setting by using an iterative procedure. Furthermore, the segmented contours are continuous and one-pixel-wide, which is another advantage of this method. However, over-segmentation problem may take place [1]. Therefore, a merge procedure is further applied in order to solve this problem.

#### 4.2.2 Connected Component Labeling

K-means clustering cannot provide a distinguished object of an image because there may be some pixels in one cluster, which are not connected to each other. Due to some disconnected pixels of a cluster, the number of regions may be more than the number clusters after the split process. In our approach, clustered image is defined by a  $m \times m$  matrix in which, pixels of each cluster have the same value with  $V_k$ . Thus, in order to label the connected pixels as a region, the connected component label algorithm has to be applied on the image. Afterwards, the initial segmented image with separated region labels is achieved. Due to noise and spatial property of an image, there will be some regions that may contain only a few pixels [13]. Small regions will be neglected if its pixel size is less than 5 pixels for an image with the

size of  $128 \times 128$ . In our approach, the size of all test images is  $256 \times 256$ . Therefore, the regions with less than 10 pixels will be ignored. In order to ignore small regions, they must be merged with one of their neighbors. Therefore, region adjacency graph (RAG) is defined to identify all the regions and their neighbors. Then, small regions can be removed by merging with their best neighbor based on some predefined criteria. The number of inappropriate regions will be reduced by removing small regions.

#### 4.3.2.3 Region Adjacency Graph

Region adjacency graph (RAG) with  $(g = \{\gamma, \varepsilon\})$  is defined as a set of nodes ( $\gamma$ ) and a set of arcs ( $\varepsilon$ ) in which, every node represents a separated region and the arcs between a couple of nodes indicate the adjacency between these two regions. An example of RAG for an initial segmented image is illustrated in Figure 4-5.

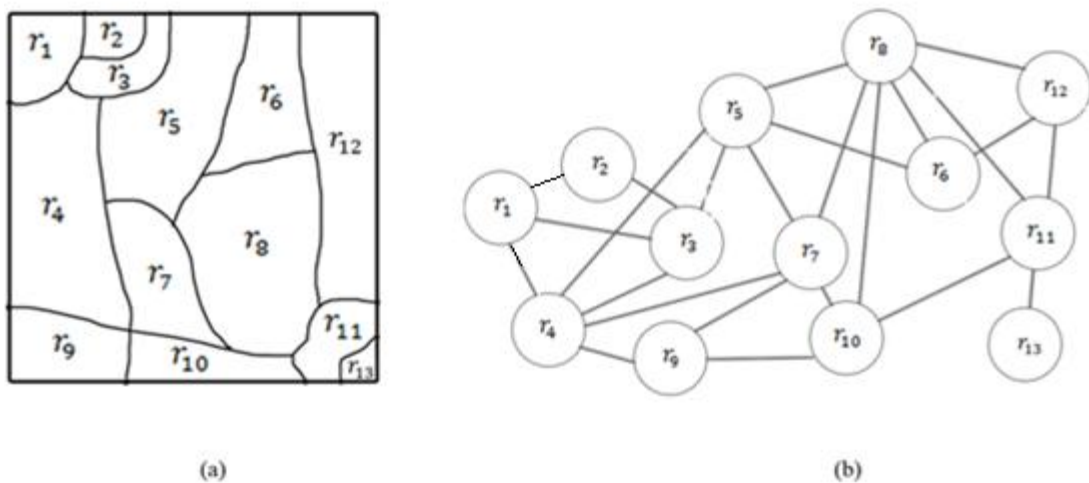


Figure 4-5.(a) Segmented Image into Separated Regions; (b) RAG of Segmented Image

The process of optimization for image segmentation is shown in Figure 4-6, in which original input image is influenced by mentioned preprocessing techniques and then detected objects are attained after the merge procedure containing a GA.

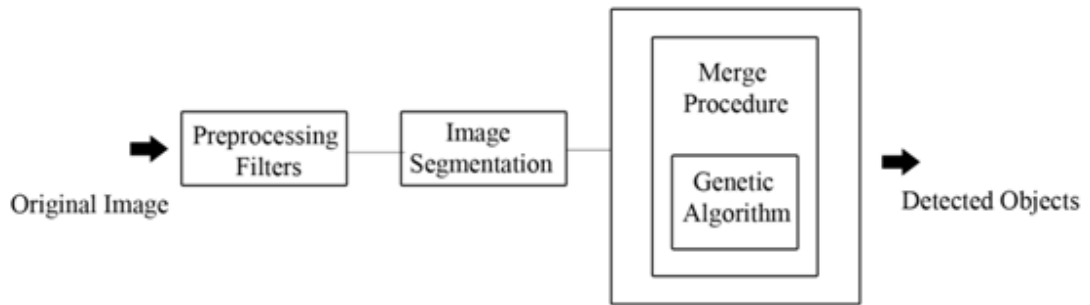


Figure 4-6. Optimization Image Segmentation Process

### 4.3 Merge Procedure Using the GAs

An evolutionary merge procedure is further applied on initial segmented image in order to ignore inappropriate regions. In this context of genetic algorithm, there are three primary states to design an evolutionary merge procedure as follows:

- (1) An effectual chromosome encoding model
- (2) A suitable fitness function
- (3) A powerful evolutionary algorithm

GA-based segmentation algorithm can be implemented as an easy iterative process in the following:

- 1) An initial population is generated from the initial split image and the fitness for each individual is computed.

- 2) Reproduction is done by mating two parents, which are selected according to their fitness values using roulette wheel selection.
- 3) The crossover operator is applied to the two parents to generate two offspring.
- 4) The mutation operator is applied in order to escape from local optimum.
- 5) The fitness values for both offspring are calculated.
- 6) Go to the step 2 until all individuals for the new population are generated.
- 7) The last population is replaced with the new population
- 8) If termination condition is satisfied, the individual with the best fitness value is extracted and the procedure stops.
- 9) If termination condition is not satisfied, go to step 2.

#### 4.3.1 Chromosome Encoding Method

A chromosome encoding method is used to represent a segmented image. Each gene of the chromosome indicates a separated region of the image. Assume that the segmented image has R regions  $(r_1, r_2, \dots, r_R)$ , let the chromosome  $\alpha = \alpha_1, \alpha_2, \dots, \alpha_N$  be an integer string, which represents the number of genes. The length of the chromosome (number of genes) is equal to the number of regions in segmented image.

In our study, chromosome structure is illustrated in Figure 4-7; each gene is defined as an agent of each region and denoted by region label  $r_R$ .

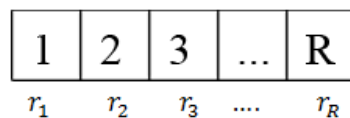


Figure 4-7. Chromosome Encoding Model of Separated Regions

### 4.3.2 Population Generation

As it is mentioned earlier, the initial segmented image is shown by a chromosome. To generate an initial population of chromosomes, a few number of genes is selected from the initial chromosome randomly and If the merging conditions are satisfied, these selected regions will be merged with one of their neighbors. Consequently, the other chromosomes (individuals) will be generated.

In each chromosome,  $\alpha_i = i$  for  $i = 1, 2, \dots, N$ , means region  $i$  should be reserved and  $\alpha_i = j$  means region  $i$  is removed by merging with region  $j$ . For instance, Figure 4-8 represents that  $\alpha_3 = 4$ , which means  $r_3$  is merged with its best neighbor ( $r_4$ ).

1	2	4	4	11	6	7	23	...	R
$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$	....	$r_R$

Figure 4-8. Initial Individual Structure

Additionally, this chromosome represents that regions 5 and 8 are merged with region 11 and 23, respectively. Other individuals are generated by repeating this merge approach.

#### 4.3.2.1 Neighbor Selection for Merge Procedure

To decide which neighbor is the best one to merge, dissimilarity distances must be compared between a region and all of its neighbors. In this study, the dissimilarity distance is based on the region variance. If the variances of two adjacent regions are close, then they are candidates for merging. Therefore, variance distances should be calculated between a region and all of its neighbors. For example, a neighbor with lower variance distance to  $r_3$  is selected as a candidate region to be merged with

region  $r_3$ . As it is shown in Figure 4-8, region  $r_3$  is to be merged with its single candidate neighbor that is  $r_4$ .

#### 4.3.2.2 Dissimilarity Distance Computation

Dissimilarity distance is calculated by measuring the Euclidean distance between variances of one region and all of its neighbors. In this approach, Euclidean distance has been computed in the scale of regions variance. As a result of the K-means clustering algorithm, each region involves a number of connected pixels ( $x$ ), with similar gray level intensity ( $g$ ).

Consequently, in order to calculate the region variance, all pixels ( $x$ ) of region  $R$  must be substituted in Equation 4-4.

$$\sigma^2_R = \frac{\sum_{i=1}^n (g_i - \bar{g})}{\bar{g}} \quad (4-4)$$

where  $\sigma^2_R$  represents variance of region  $R$ ,  $g_i$  represents the gray-level intensity of pixel  $i$  (for  $i = 1, \dots, n$ , while  $n$  is the number of pixels in the considered region) and  $\bar{g}$  indicates the mid-point of the region which is based on the gray-level intensity of pixels.  $\bar{g}$  is defined in Equation 4-5.

$$\bar{g} = \frac{\sum g_i}{N}, \text{ for } i=1, 2, \dots, N, \quad (4-5)$$

where  $N$  is the number of pixels in considering region.

To compute distance between two adjacent regions, Euclidean distance of variances is denoted by Equation 4-6.

$$d_i = \sqrt{|(\sigma^2_{r_i})^2 - (\sigma^2_{N_i})^2|} \quad (4-6)$$

where  $\sigma^2_{r_i}$  and  $\sigma^2_{N_i}$  represent variance of region  $r_i$  and variance of an adjacent neighbor of  $r_i$ .

In order to undertake the merge procedure, considering the dissimilarity distance as the only condition is not enough. Therefore, to have a desired merge procedure, a threshold value  $T$  should also be taken into account. Thus, the lower variance distance should be smaller than the predefined threshold value  $T$  to satisfy enough similarity between two adjacent regions.

Since the merge procedure causes to generate a single region from a couple of regions, RAG must be updated to prevent wrong neighbor detection for the rest of merge process. Afterwards, this approach will be repeated for the rest of candidate regions.

Description of the merge procedure is shown by the following algorithm;

```
For  $i := 0$  to  $R$ 
{
  if  $a_i \neq i$  then  $Neighbors[] := RAG(i)$ ;
  Calculate dissimilarity distance for all neighbor array members;
   $Min\_distance :=$  select neighbor with lower distance;
  If  $Min\_distance < T$ ;
  { change region label of two regions into a same label;
    remove boundary between regions with the same label;
    update RAG;
  }
}
```

where  $a_i$  represents the value of  $i^{th}$  gene of the chromosome and  $T$  is a predefined threshold value.

### 4.3.3 Fitness Function

A variety of objective functions of image segmentation algorithm have been proposed for different objectives and applications. However, it is not easy to design a universal objective function for different applications [38]. In this representation, the K-means algorithm is based on the properties of the pixels in a region. Consequently, the fitness function is also related to the overall properties of the regions in an image. We can define the fitness function with two main objectives, namely: regions contrast (dissimilarity distances between regions) and region size. Therefore, the fitness function for each chromosome is based on the summation of all variance distances between the merged regions. Fitness value is computed to evaluate the quality of each individual. Based on the predefined fitness function, better fitness values identify better individuals.

For example, the candidate regions, which are supposed to be merged, are  $r_3, r_5, r_8, r_{24}, r_{43}$ . As it is mentioned previously, the minimum dissimilarity



distance ( $d$ ) between each of these regions and their neighbors should be found. The summation of these distances must be calculated to determine the fitness function. In addition, the region size must be substituted in fitness function. For these reason, we can define the fitness function as in Equation 4-7.

$$F = \sum_{i=1}^N \frac{1}{d_i * s_i}, i = 1, \dots, N \quad (4-7)$$

where  $i$  represents the candidate regions which should be merged with their best neighbors,  $s_i$  indicates the size of removed region and  $d_i$  represents the Euclidean distances between the removed regions and their closest neighbor based on variance.

The smaller value of  $s_i$  and  $d_i$  produce better fitness value. In our problem the larger value of fitness function is more desirable. As a result, we used multiplication of these two values as denominator to have higher fitness value.

#### **4.3.4 Selection Operator in GAs**

To select the best pair of chromosomes, the fitness value is an essential parameter to generate the next population. In this study, a selection operator is applied based on the Roulette Wheel Selection (RWS) [39] approach. In RWS approach, each individual occupies a portion of roulette wheel, with respect to its fitness value. Individuals with better fitness values take longer slot from the roulette wheel. It is clear that the longer slot has the higher priority to be selected. In our research, two chromosomes are selected. Consequently, to produce new chromosomes, crossover and mutation operations are applied.

#### 4.3.4.1 Roulette Wheel Selection

In this approach, fitter individuals will tend to have a better chance of survival and will go forward to produce the mating pool for the next generation. Although there is a high chance for selecting the fitter individuals, there is still a chance for weaker individuals to be selected. In nature such individuals may have genetic coding that may prove useful to future generations. This common selection approach assigns a proportion of selection  $P_j$  of selecting each individual  $j$  based on its fitness value. The proportion  $P_i$  for each individual is defined by equation 4-8, [39].

$$P_i = \frac{F_i}{\sum_{j=1}^{N_p} F_j} \quad (4-8)$$

where  $F_i$  represent the fitness value of individual  $i$  and  $N_p$  indicates the population size.

Fitness function values of all individuals from each population are computed as  $F_{I_1}, F_{I_2}, \dots, F_{I_n}$ . Each of these fitness values occupies a specific percentage of a roulette wheel. The better fitness value takes the higher portion of the roulette wheel. The roulette-wheel is spun  $n$  times. Since the circumference of the wheel is marked according to a string of fitness values, this roulette-wheel mechanism is expected to make  $F_i/\bar{F}$  copies of the  $i^{th}$  string in the mating pool.

The  $\bar{F}$  is calculated by Equation 4-9, [39].

$$\bar{F} = \sum_i^N F_i \quad (4-9)$$

where  $N$  represents the number of individuals and  $F_i$  is the fitness value of  $i^{th}$  individual.

Figure 4-9 shows an example of roulette-wheel in which each individual has different fitness values. Since the third individual has the highest fitness value, it is more likely that the roulette-wheel selection choose the third individual more than the other individuals.

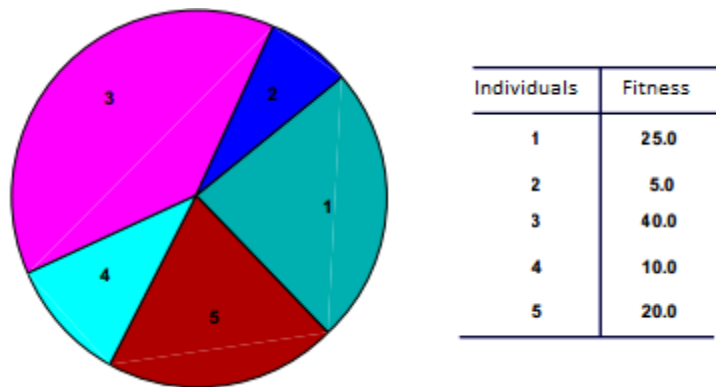


Figure 4-9. A Roulette Wheel Marked for Five Individuals According to Their Fitness

Values [39]

Pseudo code of roulette wheel selection is illustrated as follows:

```
BEGIN
  Set current_member =1
  WHILE (current_member ≤  $N_p$ ) DO
    Pick a random value r uniformly from [0, 1];
    Set i=1;
    Set Sum=  $a_i$ ;
    WHILE Sum < r DO
      Set i= i+1;
      Sum= Sum+  $a_i$ ;
    OD
    Set mating_pool [current_member]= i;
    Set current_member= current_member+1;
  OD
END
```

where  $N_p$  represents population size and  $a_i$  is the fitness proportion of  $i^{th}$  individual.

#### 4.3.5 Crossover Operation

In this implementation, two-point crossover operation is applied on the selected chromosomes (parents), to produce two offspring. Through the crossover operation, a fixed interval with a random selected start point is defined as the crossover interval. The length of crossover interval depends on the number of separated regions in the segmented image. The entire genes of the crossover interval will be swapped between two parents, hence generating two offspring. This transformation manipulates the feasibility of the both chromosomes. Therefore, in order to sustain the feasibility, genes with the wrong value should be updated.

If the crossover segment of a parent chromosome includes some merged regions with the other regions out of the crossover interval, then existence of a relation

between these regions will be approved. This relation must be maintained through the crossover operation. As mentioned above, in order to generate new chromosomes, the selected genes must be swapped between two parents. Then those genes which are out of crossover interval and depending on crossover interval must be updated by means of applying the relation between gens.

For instance, two individuals are selected and illustrated in Figure 4-10. The chromosome has 15 genes in length, which means the image is segmented into 15 regions.

Parent 1	1	2	4	4	11	2	7	15	9	12	11	12	13	1	15
	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$	$r_9$	$r_{10}$	$r_{11}$	$r_{12}$	$r_{13}$	$r_{14}$	$r_{15}$

Parent 2	3	3	3	4	8	6	3	8	14	11	11	14	13	14	4
	$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$	$r_9$	$r_{10}$	$r_{11}$	$r_{12}$	$r_{13}$	$r_{14}$	$r_{15}$

Figure 4-10. Two Selected Chromosome as the Parents

Then length of crossover segment is defined (according to the ratio of parent length) by the Equation 4-10.

$$crossover\ segment\ length = round\left(\frac{L_c}{C}\right) \quad (4-10)$$

where  $L_c$  indicates the chromosome length,  $C$  represents an arbitrary constant.

If  $C$  is very big, then a few numbers of genes will be selected for crossover operator. As a result, the rate of convergence will be very slow. However, by using very small  $C$ , more genes will participate in crossover operation. Therefore, diversity of the population will become very low. After comparing the results of this approach with different values of  $C$ ,  $C = 3$  produced better results.

For instance, in Figure 4-11, crossover segment is presented in the range of 5 genes, which is started from  $r_6$ .

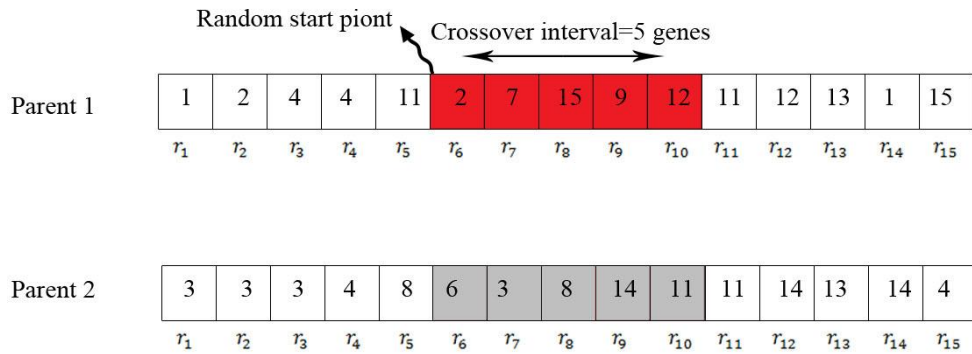


Figure 4-11 An Example of the Crossover Segment on Parents.

Crossover segment of parent 1 contains 5 genes in which  $r_6, r_8$  and  $r_{10}$  are merged with  $r_2, r_{15}$  and  $r_{12}$  respectively. As it is mentioned earlier,  $\alpha_6 = 2$  means that  $r_6$  is removed by merging with  $r_2$ . As a result, we should have  $\alpha_2 = 2$  in order to have feasible chromosome, which means  $r_2$  is a reserved region. Besides, in the crossover interval of parent 2,  $r_7, r_9$  and  $r_{10}$  are merged with  $r_3, r_{14}$  and  $r_{11}$ , respectively.

Afterwards, genes in crossover segment of both parents are swapped as indicated in Figure 4-12.

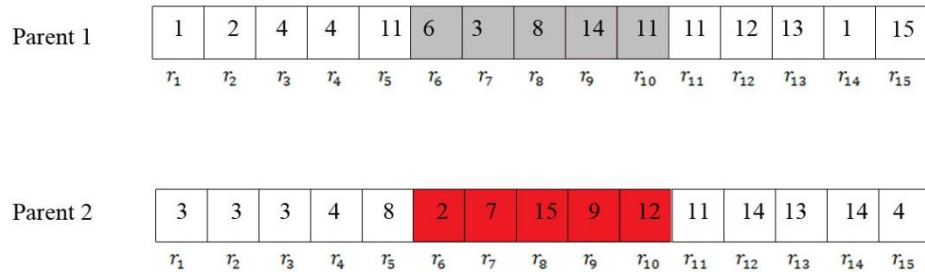


Figure 4-12. Swapped Crossover Segment between Two Parents.

Then, two offspring are generated by means of swapping the genes. In the crossover interval of offspring 1,  $r_7, r_9$  and  $r_{10}$  are supposed to be merged with  $r_3, r_{14}$  and  $r_{11}$ , respectively. As a result,  $r_3, r_{14}$  and  $r_{11}$  should be reserved. It means that the condition  $\alpha_3 = 3$ ,  $\alpha_{14} = 14$  and  $\alpha_{11} = 11$  must be satisfied. However, in offspring 1,  $\alpha_3 = 4$  and  $\alpha_{14} = 1$  are required to be updated to prevent infeasibility. The result of the crossover operation is illustrated in Figure 4-13.

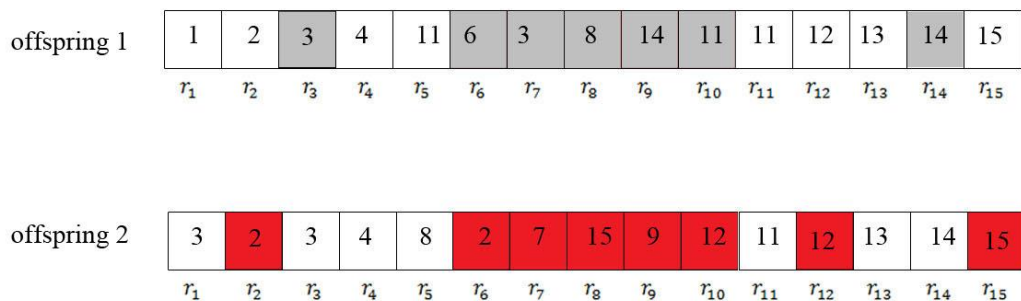


Figure 4-13: Representation of the Crossover Operator

### 4.3.6 Mutation Operation

In our representation, mutation operation causes to change the region label into one of its neighbors. In this method, random numbers of genes (regions) are selected from the offspring. The selected regions are going to be merged with one of the adjacent neighbors as an extra merge procedure. As a result of this method, in order to produce one uniform region, the boundaries between two merged regions will be removed. For example, two regions ( $r_1$  and  $r_{13}$ ) from offspring 1, are selected for mutation operation and then merged with  $r_3$  and  $r_6$ , respectively. Besides, region  $r_4$  from offspring 2 is merged with region  $r_{14}$ . A representation of mutation operator is illustrated in Figure 4-14.

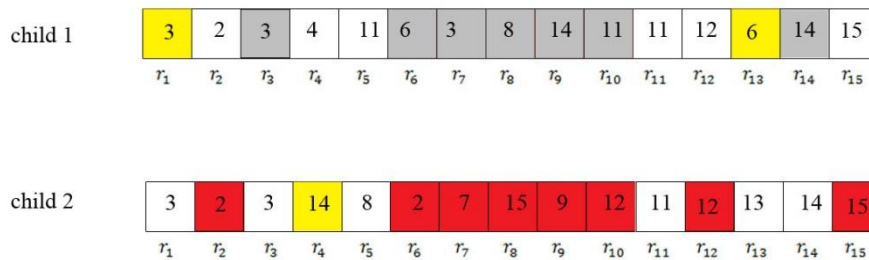


Figure 4-14. Results of the Mutation Operation

### 4.3.7 Elitism Strategy

In order to take more advantages from GA optimization technique, the elitism strategy is used during the generation approach. In canonical GA, in order to produce the next population, all of the individuals from the current population must be participated. Therefore, the entire individuals from previous population will be replaced by the new generation [34]. Thus, the convergence rate of canonical GA is found to be very low. On the other hand, in elitism strategy, the best individual from



the current population always survived into the next population [34]. As a result, CPU time is lower and the individual with the best fitness value would be unchanged in the next generations.

As an example, for the above described steps, a particular instance is taken and the computations are illustrated step by step. Let us consider the chromosome encoding model of an initial segmented image as illustrated in Figure 4-15.

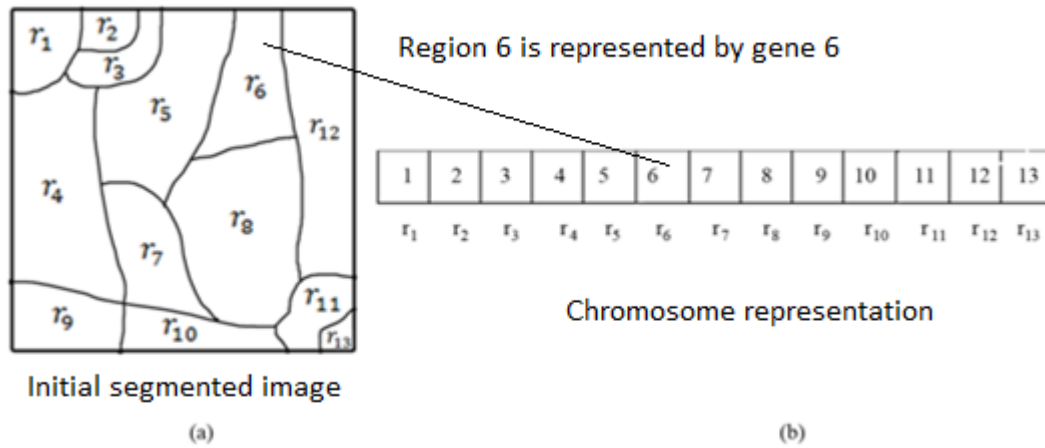


Figure 4-15. An Example of Chromosome Representation of Segmented Image

In order to generate the heuristic initial population, some of genes of the initial chromosome are selected to be merged with their closest candidate neighbor. For example, supposed that candidate regions are  $r_3, r_5, r_8, r_{13}$  by using Equation 4-9 with  $c=3$  and  $l_c=13$ . It means these candidate regions must be merge with one of their adjacent neighbors.

To find the closest neighbor of the  $i^{th}$  region, following algorithm should be considered.

```

N[ ] = neighbors of candidate_regions [i];
For j := 1 to length(N)
    Euclidean_distance[j] :=  $\sqrt{|\sigma^2_i - \sigma^2_{N[j]}|}$ ;
End
[d, Min_Index] = min[ Euclidean_distance[j] ]
Return Min_Index;

```

Finally, *Min\_Index* represents the closest neighbor of the first candidate region. Consequently, these two regions will be merged together. By following the previously mentioned algorithm, all neighbors of the candidate regions are extracted from the below subsets of RAG in Figure 4-16.

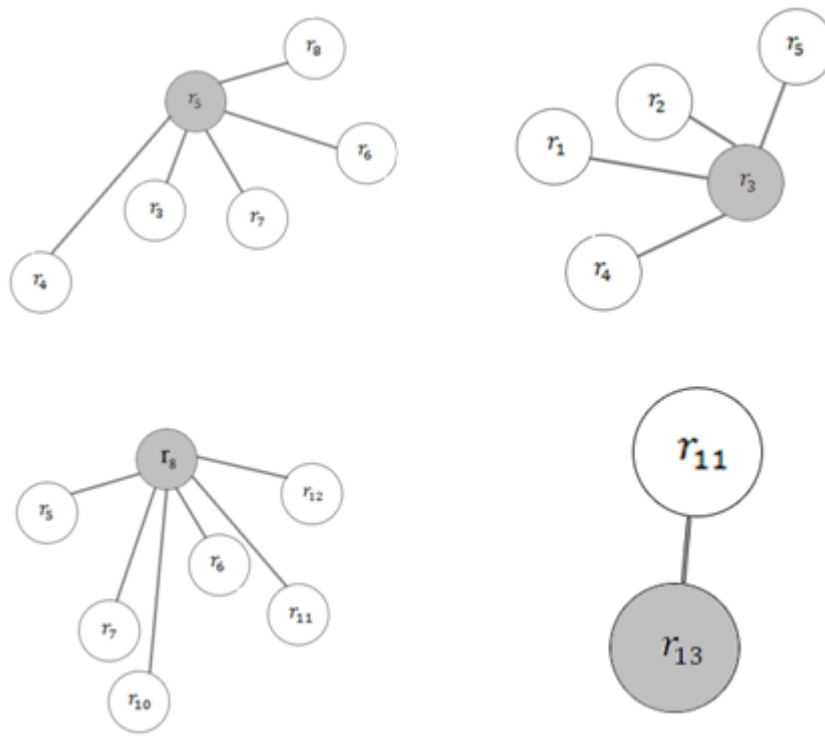


Figure 4-16. The Subsets of RAG

Candidate regions are not merged with one of their neighbors randomly. Two predefined conditions must be satisfied namely: minimum dissimilarity distance and threshold technique.

Then the Euclidean distances of variance will be computed between each candidate region and all of its neighbors. For example, the minimum distance is assumed to be 1.05, which belongs to  $r_3$  and  $r_4$ . In addition, this minimum distance must be smaller than a predefined threshold  $T$ . Therefore,  $r_3$  is removed from segmented image by merging with  $r_4$  in Figure 4-17.

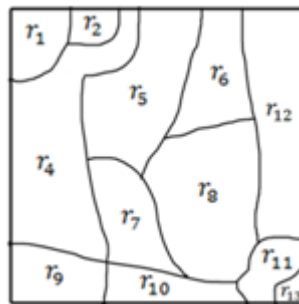


Figure 4-17. Merge Procedure in Segmented Image:  $r_3$  and  $r_4$  are Merged

After this merge procedure, RAG must be updated and  $r_3$  should be changed to  $r_4$  in the other subsets of RAG. Consequently, the neighbors of  $r_5$  will be changed in Figure 4-18.

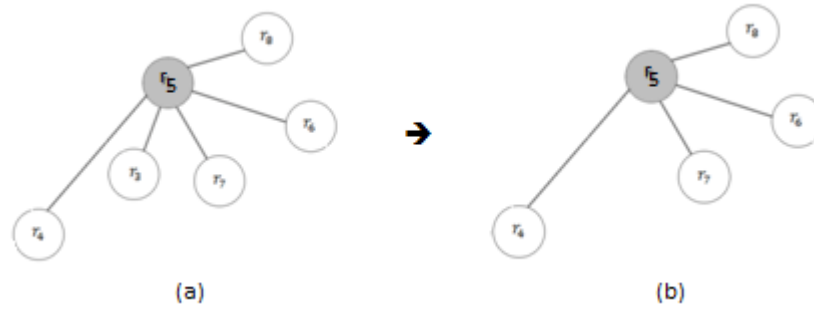


Figure 4-18. Subsets of RAG: (a): Neighborhood of  $r_5$  before Merge Procedure; (b): Neighborhood of  $r_3$  after Merge Procedure by Updating the RAG

When the merge procedure for  $r_3$  is completed, the rest of candidate regions are considered to be merged with one of their neighbors. For example,  $r_6$  has the lowest distance with  $r_5$  by 0.98 and it is smaller than  $T$ . Therefore,  $r_5$  is removed from the segmented image by merging with  $r_6$  in Figure 4-19.

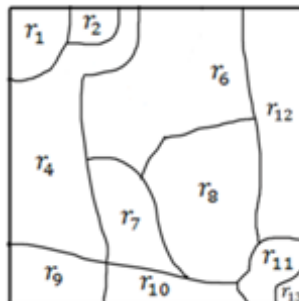


Figure 4-19. Merge Procedure in Segmented Image:  $r_5$  and  $r_6$  are Merged

When this procedure is completed, the RAG needs to be updated again. Accordingly,  $r_5$  is change to  $r_6$  in the following subsets of the RAG in Figure 4-20.

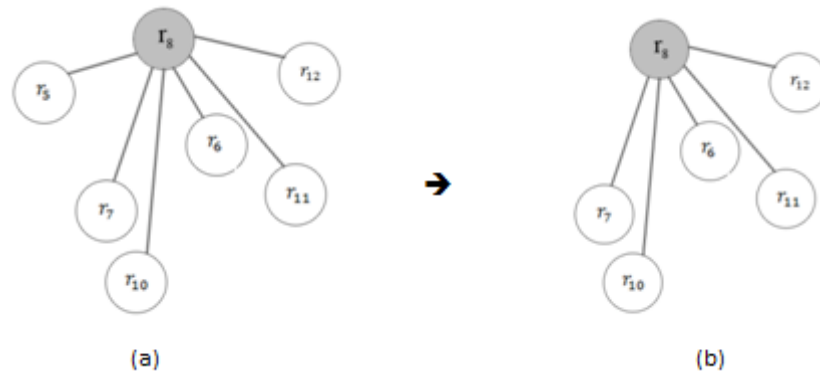


Figure 4-20: Subsets of RAG: (a): Neighborhood of  $r_8$  before Merge Procedure; (b): Neighborhood of  $r_8$  after Merge Procedure by Updating the RAG.

In this way, we assume that  $r_{12}$  has the lowest distance with  $r_8$  by 4.08. However, this minimum distance is not smaller than  $T$ . Therefore, necessary conditions for merge procedure are not satisfied. Afterwards,  $r_{13}$  has the lowest distance with  $r_{11}$  by 1.88, which is smaller than  $T$ . thus,  $r_{13}$  will be merged with  $r_{11}$ . This mere result is illustrated in Figure 4-21.

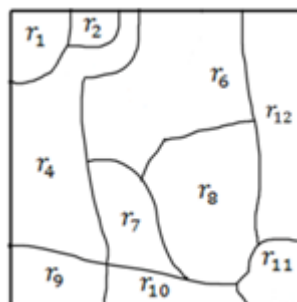


Figure 4-21. Merge Procedure in Segmented Image

When the merge procedure is completed, a new chromosome will be generated as indicated in Figure 4-22.

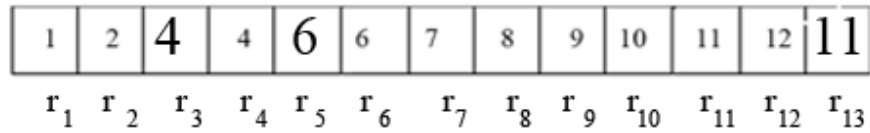


Figure 4-22. New Chromosome Representation after Merge Procedure

This chromosome is defined as an individual of the first population. The initial population will be completed with 10 different members, by repeating this process for 10 times (in this example population size is 10).

To compute the fitness value of each chromosome, the minimum distance between candidate regions and their best neighbors are substituted in the equation of fitness function. Assuming that minimum for  $i= 3, 5, 13$  are 1.05, 0.98, 1.88 and sizes for  $i= 3, 5, 13$  are 25, 56, 16, the corresponding fitness value is:

$$F = \sum_{i=1}^n \frac{1}{d_i * s_i}, i = 3, 5, 13$$

$$= \frac{1}{1.05 * 25} + \frac{1}{0.98 * 56} + \frac{1}{1.88 * 16} = 0.000381 + 0.0001822 + 0.0332447 = 0.0338079$$

As a result of this computation, the fitness value of the initial chromosome is 0.0338079.

In the Figure 4-23, 10 different chromosomes of initial population are illustrated and the fitness value is computed for all individuals.

1	2	4	4	6	6	7	8	9	10	11	12	13	F=0.0338079
2	2	3	4	5	6	10	8	9	10	11	12	11	F=0.03965
1	1	1	4	5	6	7	8	9	9	11	12	13	F=0.0429805
4	4	4	4	5	6	7	8	10	10	11	12	13	F=0.051027
1	3	3	4	5	6	7	8	9	10	13	12	13	F=0.02822
2	2	2	4	6	6	7	8	9	10	11	12	13	F=0.009826
1	2	3	3	5	6	9	8	9	10	11	12	13	F=0.059
1	2	3	4	4	8	7	8	9	10	11	11	11	F=0.046902
1	1	1	1	5	6	7	8	9	10	11	12	13	F=0.063072
1	2	3	3	5	5	7	8	10	10	11	12	13	F=0.033726

Figure 4-23. An Example of Initial Population Includes 10 Different Individuals and Their Computed Fitness Values

The best chromosome is the one which has the highest fitness value among the other individuals. In this initial population, chromosome with fitness value equal to 0.063072 is the best chromosome.

## Chapter 5

### RESULTS AND DISCUSSION

The proposed genetic algorithm optimization for image segmentation (GOIS) is implemented using Matlab programming on an Intel core (TM) i3CPU computer. The procedure will be completed when the fitness value of the best chromosome remains unchanged during 5 generations. Two experiments are done to determine the efficiency and strength of GOIS. Experiment 1 used one image of size  $256 \times 256$  which is corrupted using Gaussian noise to illustrate the power of GOIS under noisy conditions. Experiment 2 used five images of size  $256 \times 256$  with diverse characteristics to show the efficiency of GOIS.

- (1) Lena; Figure 5-1 (a): This image is usually used in image processing applications. Clear background and numerous fragmental regions are the remarkable reasons of using this image.
- (2) Yacht; Figure 5-8 (a): Due to the reflection of water, which causes to produce different gray-level intensity, it is difficult to segment this image and then merge a variety number of inappropriate regions.
- (3) Fruit; Figure 5-9 (a): This image includes 7 fruits in which over-segmentation problem for each fruit may occur because of the reflection.
- (4) Husky; Figure 5-12 (b): This image contains discrete dissimilarity between background and the Husky. The characteristic of this image is that the outline



of Husky's hair is unclear. It is difficult to distinguish Husky from its background.

(5) Lake; Figure 5-12 (c): This image contains abundant leaves which have diverse gray-level value. It is difficult to merge the various leaves of a tree into a single region.

## 5.1 Evaluation of GOIS Approach on Noisy Images

In order to prove the power of GOIS approach in noisy image, original input image is corrupted by Gaussian noise. Image of Lena is used for this experimental part. Gaussian noise is statistical signal that has its probability density function equal to that of the normal distribution, which is also known as the Gaussian distribution. A special case is white Gaussian noise, in which the values at any pairs of times are statistically independent (and uncorrelated). In applications, Gaussian noise is most commonly used as additive white noise to yield additive white Gaussian noise. In the Gaussian noise, the intensity value of the pixel  $x(i,j)$  is altered by additive white Gaussian noise  $x(i,j)+n$ , with the noise  $n \sim N(\nu, mean)$  being normally distributed with variance  $\nu$  specified by the user. As a result, after applying Gaussian noise on the pixels of image, the intensity values of the pixels are influenced by the values of mean and variance.

First, K-means algorithm is applied on Lena image. Then the split results are illustrated in Figure 5-1. Secondly, K-means algorithm is applied on corrupted Lena image by Gaussian noise (variance= 0.001 and mean =0.0) and the segmented outcomes are shown in Figure 5-2. Finally, K-means algorithm is applied on noisy Lena image with Gaussian noise (variance= 0.05 and mean= 0.0) and the segmented outcomes are shown in Figure 5-3.

Figure 5-1, represents the result of K-means algorithm on gray-level intensity image. As a result of this clustering technique, there are 4 clusters for the entire pixels of the image. However, there are more than 4 regions due to existence of numerous discrete clusters.

### 5.1.1 Split Procedure Results

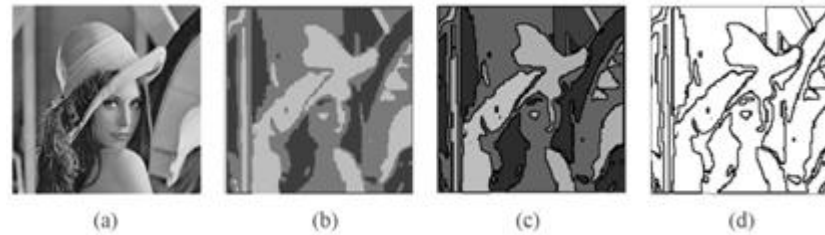


Figure 5-1. Test Image: (a) Original Gray Lena Image; (b) Clustered Image Using K-means Algorithm; (c) Separating Image by Drawing Boundary; (D) Region Detection

As it is illustrated by color of clusters in Figure 5-1 (b), Lena's hat and her shoulder and some sporadic parts of the image belong to one cluster. On the other hand, these parts indicate separated regions due to discontinuous clusters.

However, adding noise to the image results in over-segmentation by using K-means clustering algorithm. In the following, Figures 5-2 and 5-3 explain two different ranges of Gaussian noise resulting in corrupted Lena images.



Figure 5-2. Test Image: (a) Corrupted Image with Gaussian Noise (Variance= 0.001 and Mean= 0.0); (B) Clustered Image Using K-Means Algorithm; (C) Separating Image by Drawing Boundary; (D) Region Detection

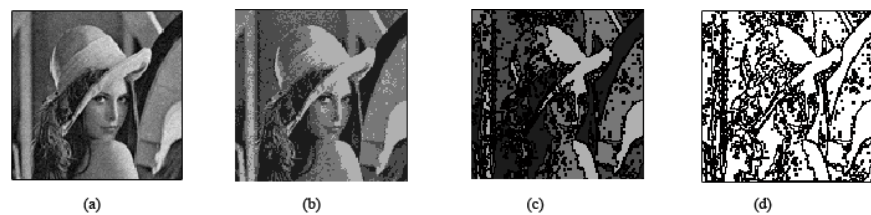


Figure 5-3. Test Image: (A) Corrupted Image with Gaussian Noise (Variance= 0.05 and Mean= 0.0); (B) Clustered Image Using K-Means Algorithm; (C) Separating Image by Drawing Boundary; (D) Region Detection

Due to poor localization and over-segmentation, the split results of these two noisy images are degraded. However, this problem will be solved by our evolutionary optimization approach. When the split process is finished, to prevent the over-segmentation, removing the small regions approach is further defined as a filter on the first segmented image, which this process has to be done before merge procedure.

Experimental split results on clustering and preprocessing of the above mentioned Lena images are shown in Table 5-1.

Table 5-1. Split Results of Three Different Noises on Lena Image

	Number of regions after applying K-means	Number of regions after removing small regions	Duration of K-means (s)	Duration of The Preprocessing (s)
<b>Figure 5-1(d)</b>	45	39	15	32
<b>Figure 5-2(d)</b>	338	182	20	30
<b>Figure 5-3(d)</b>	569	213	27	55

It can be said from Table 5-1 that images with higher level of noise suffer from over-segmentation problem due to existence of many inappropriate regions. Due to participating noise in K-means algorithm, which has the same priority with other pixels, time consumption is almost independent of noise. On the other hand, removing small regions takes much time because of the existence of more small regions in the image with higher level of noise. Chromosome's size will be determined after applying small region remover filter on the image. In such way, number of regions of the initial segmented image will be decreased by the number of small regions.

### 5.1.2 Merge Procedure Results Based on GA Optimization

In order to optimize the segmentation results, which is based on K-means algorithm, a GA is defined for merge procedure. In this representation, there are 25 individuals in each population ( $N_p=25$ ). The quality of each individual is measured by the fitness values through the generations. In this experiment, an individual with the best fitness value is selected as the best one among all individuals. Therefore, the best individual of each population is illustrated in Figures 5-4, 5-5 and 5-6 through the merge

procedure with various numbers of generations. Merge procedure will be terminated when the fitness values remains unchanged over 5 generations.

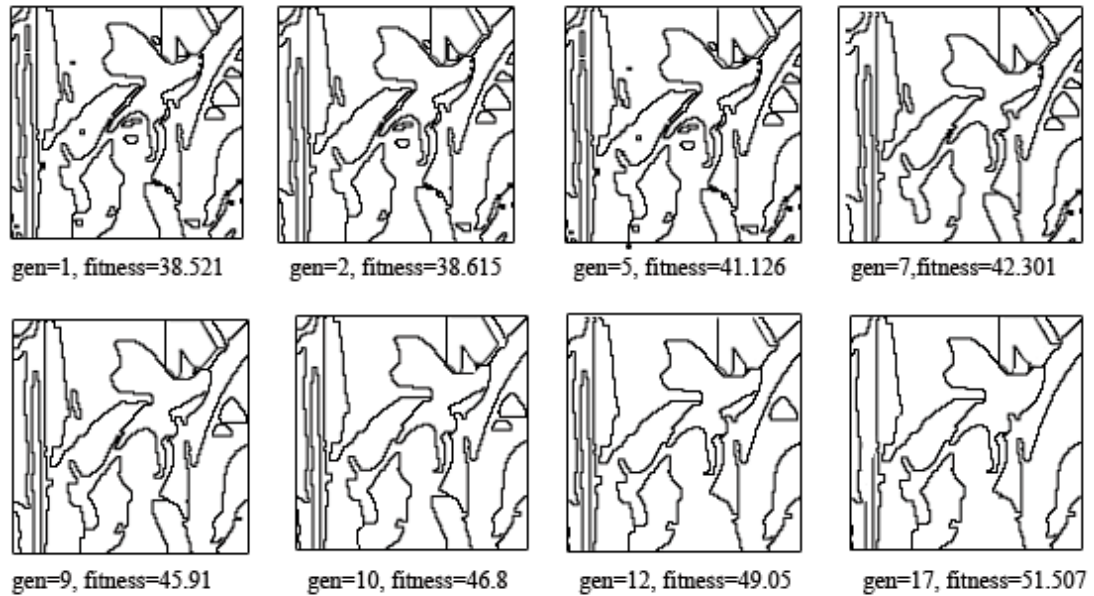


Figure 5-4. Merge Procedure of Original Lena Image after Various Numbers of Generations

In the next section of our implementation, two different levels of Gaussian noise are applied to Lena image and then merge procedure is performed with the various numbers of generations in Figures 5-5 and 5-6.

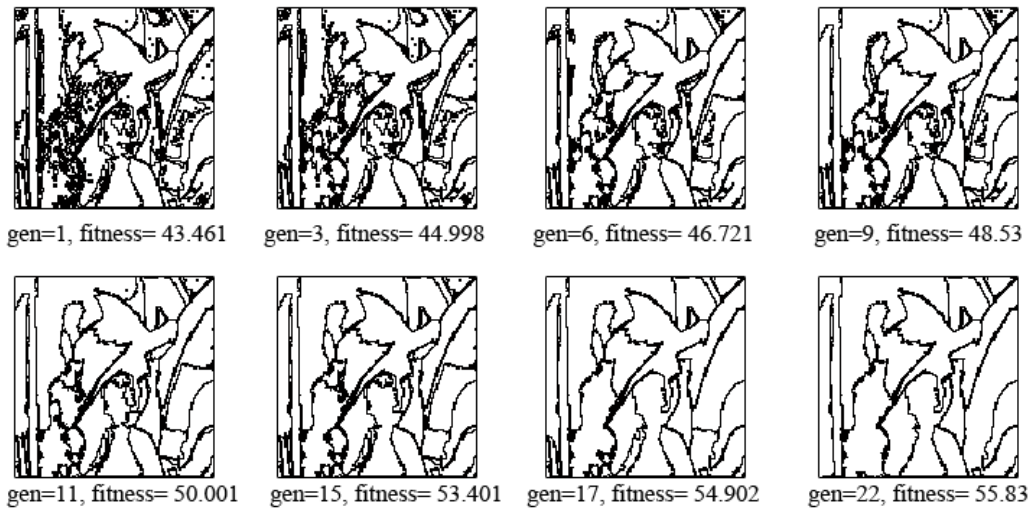


Figure 5-5. Merge Procedure of Corrupted Lena Image by Gaussian Noise (variance= 0.001 and mean=0) after Various Number of Generations

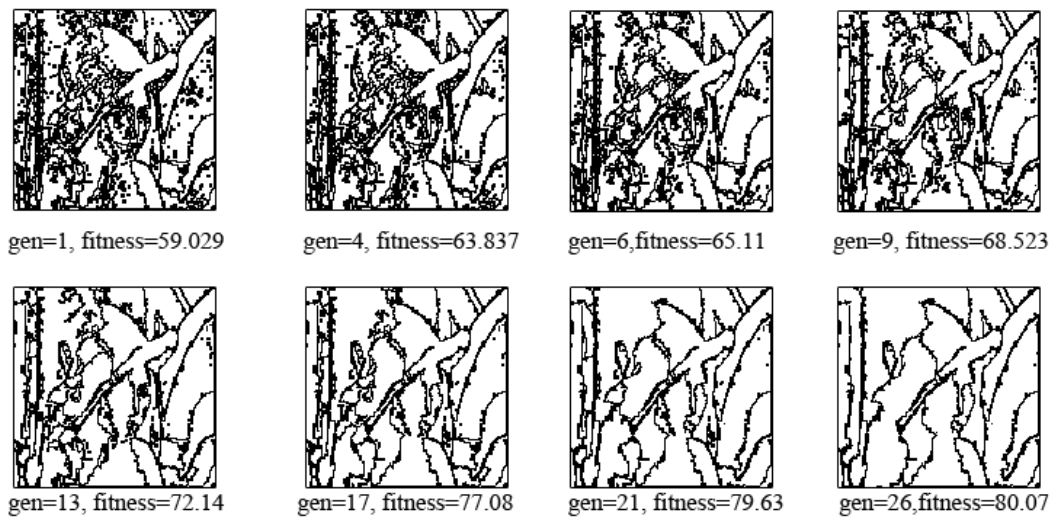


Figure 5-6. Procedure of Corrupted Lena Image by Gaussian Noise (variance= 0.05 and mean=0) after Various Number of Generations

A graphical representation of three different experimental results (Figures 5-4, 5-5 and 5-6) are illustrated in Figure 5-7, based on their fitness values.

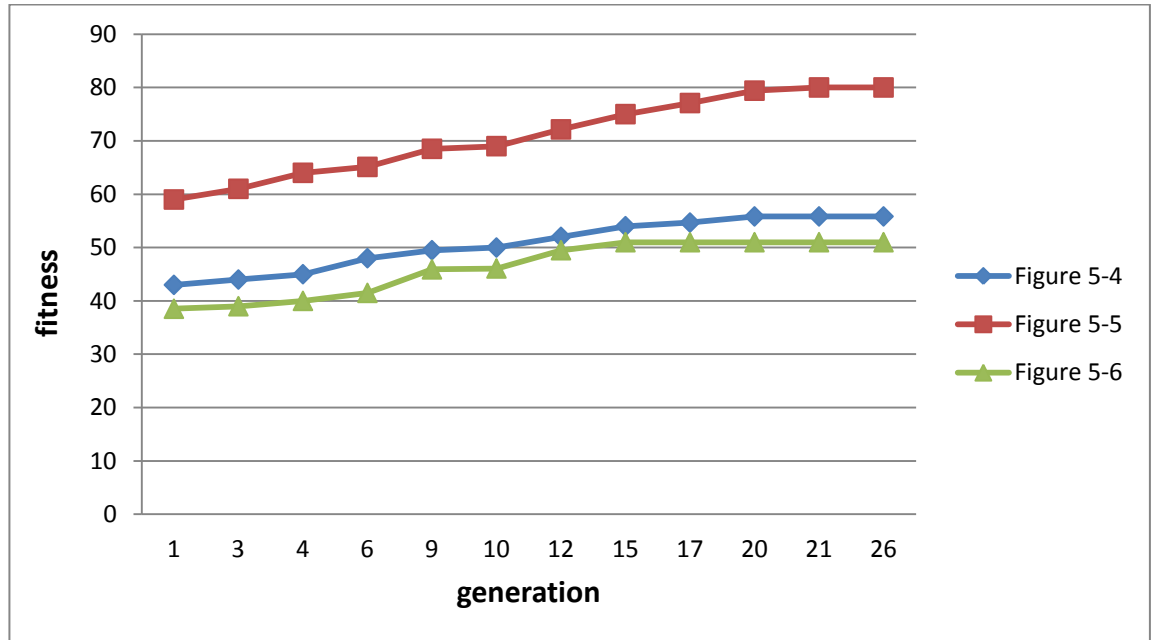


Figure 5-7. Comparison of the Fitness Values through the Various Generations

As it is indicated in Figure 5-7, the noisy images reached to a desired segmentation result in comparison with the reduced noise image after longer generations.

Comparison of the CPU time consumptions for these three Lena images has been tabulated in Table 5-2.

Table 5-2. Comparison of CPU Times for the Three Different Lena Images Resulting of GOIS

Lena image	Number of generations	Chromosome length	Number of detected objects	Termination time
<b>Figure 5-4</b>	17	39	15	97.01
<b>Figure 5-5</b>	22	182	19	130
<b>Figure 5-6</b>	26	213	23	169.55

As a result of these experiments, it is obvious from Table 5-2, that GOIS can be also efficient for noisy images in the context of image segmentation. However, noisy images require longer generations and CPU times for optimization.

## **5.2 Evaluation of GOIS Approach on Images with Different Characteristics**

In this experiment, two source images (Yacht and Fruit) are used as input to GOIS algorithm. Then the split results of these images will be compared. Afterwards, the segmentation results of GOIS are compared with three other segmentation algorithms, namely: Canny operator mask [27], simple genetic algorithm (SGA) [31], and FCM [13].



- (1) SGA is a typical genetic algorithm which is a flexible problem-solving technique. In this algorithm, one-point crossover and simple mutation operations are applied on chromosomes [1].
- (2) Canny operator mask is typically used for edge detection based on predefined threshold  $T$ .
- (3) FCM can be used for image segmentation based on Fuzzy pixel classification [1]. It results in higher flexibility in image processing [13]. In experiments, the cluster number of FCM is equal to 2.

### 5.2.1 Split Procedure Results

Image Yacht and image Fruits are segmented using K-means clustering and then merge procedure has been applied during 15 and 20 generations respectively. In Figures 5-8 and 5-9, it can be seen that various numbers of inappropriate regions are merged with their best neighbors.

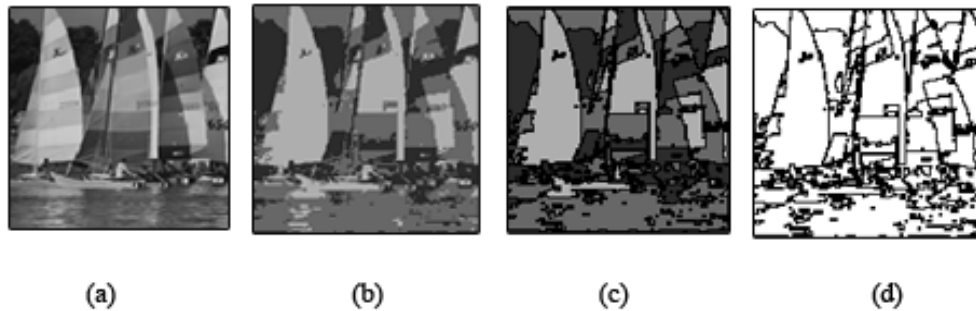


Figure 5-8. Test Image: (a) Original Gray-Level Image; (b) Clustered Image Using K-Means Algorithm; (c) Separating Image by Drawing Boundary; (d) Region Detection

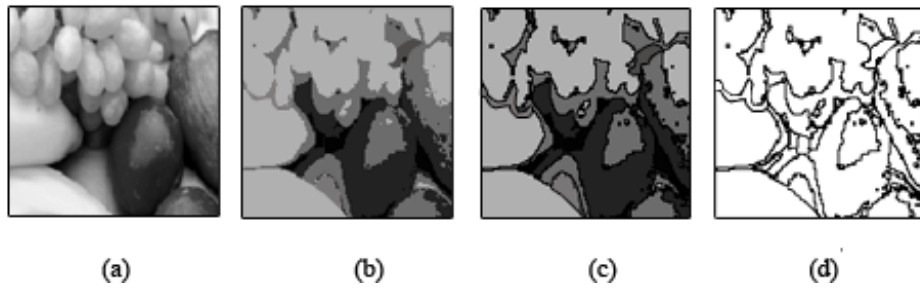


Figure 5-9.(a) Original Gray Fruit Image; (b) Clustered Image Using K-Means Algorithm; (c) Separating Image by Drawing Boundary; (d) Region Detection

Table 5-3 illustrates the results associated with these three images: Lena, Yacht and fruits.

Table 5-3.Comparison of Three Images Resulting from the Split Procedure

	Number of regions after using K-means	Number of regions after removing small regions	Duration of K-means (s)	Duration of preprocessing (s)
<b>Lena</b>	45	39	15	32
<b>Fruit</b>	61	19	17	42.09
<b>Yacht</b>	139	79	45	60.2

Merge procedure is further applied on split results through the GA with various number of generations as shown in Figures 5-10 and 5-11.

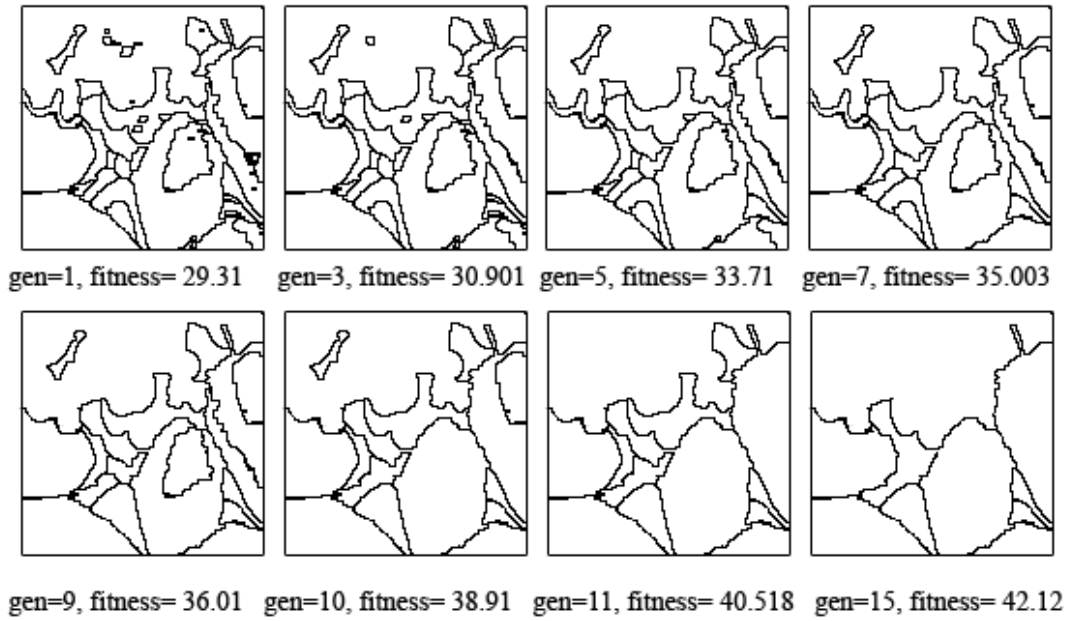


Figure 5-10. Merge Procedure of Original Fruit Image after Various Numbers of Generations

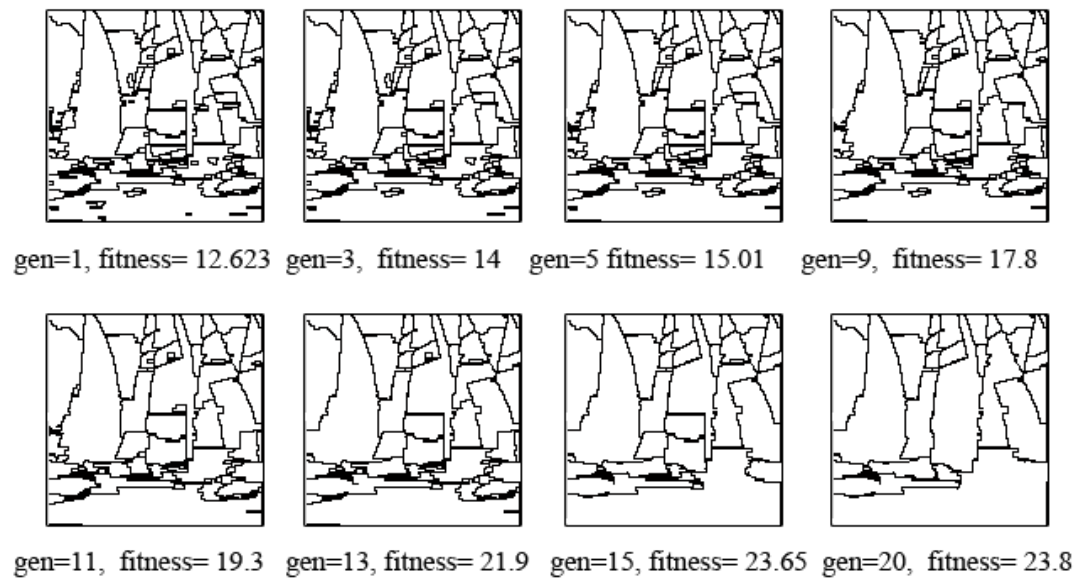


Figure 5-11. Merge Procedure of Original Yacht Image after Various Numbers of Generations

By comparing the merge approaches for these two different images, it can be said that the number of generations to attain desired segmentation for Yacht image is

more than Fruit image due to different characteristics of these images. Yacht image consists of various intensity levels in water, which produces over-segmentation in the split image. As a result of over-segmentation, chromosome size is longer and more genes are considered in merge procedure. Therefore, Yacht image needs 20 generations to remove inappropriate regions and achieve to the desired segmentation. However, 15 generations are enough for Fruit image to complete merge procedure through the segmentation.

Comparison of split results for the three different images is shown in the following Table 5-4.

Table 5-4. Comparison of Three Different Images Resulting of Merge Procedure

	<b>Number of generations</b>	<b>Chromosome length</b>	<b>Detected object numbers</b>	<b>Termination time (s)</b>
<b>Lena</b>	17	39	18	97.01
<b>Fruit</b>	15	19	8	101.73
<b>Yacht</b>	20	79	28	194

As it is indicated in Table 5-4, longer chromosomes require higher computation time due to existence of many genes, which need to be participated in crossover and mutation operators. In addition to chromosome length, more number of generations causes to increase the termination time.

### 5.3 Comparison of GOIS Approach with Other Approaches

In the last section of our experiments using GOIS, three images are compared with other segmentation methods in Figure 5-12.

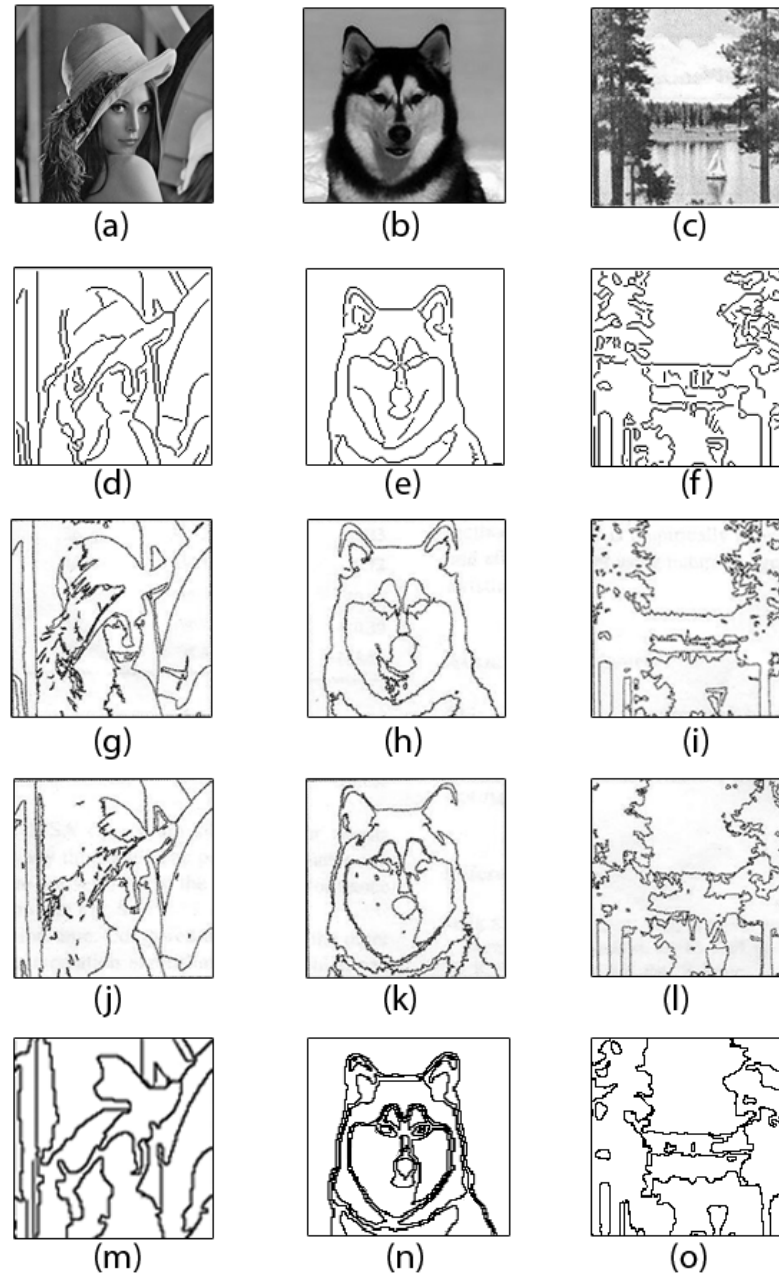


Figure 5-12. Comparisons of Four Image Segmentation Approaches: (a)-(c) Source Images with the Size of  $256 \times 256$ ; (d)-(f) Results of Canny Segmentation; (g)-(i) Results of FCM in [1]; (j)-(l) Results of SGA [1]; (m)-(o) Results of GOIS

As it is obvious from the experimental results shown in Figure 5-12, the split results of GOIS approach are much better than the other experimental results in this study. Canny segmentation approach (d-f) is fast but in some parts of these results the contours are not continuous. Therefore, Husky image is not detected as a close object and in some parts of Husky's ear, the boundary is discontinuous. On the other hand, the results of GOIS approach illustrate accurate edge detection for each object of the image.

The results of FCM approach [1] have poor localization on the boundaries and also discontinuous detected edges. Also, there were some inappropriate regions due to over-segmentation. In comparison with FCM, GOIS approach can prevent over-segmentation through the number of generations in merge procedure.

#### **5.4 Population Size**

Large population size ( $N_p$ ) may result in a perfect final performance. However, it may result in high computation time. Typical values of  $N_p$  is in the interval [20, 100], [1]. In this study, after comparing the results of GOIS approach with different  $N_p$ , the GA's population with  $N_p=25$  produced better results. To compare the results of GOIS approach with different  $N_p$ , three images are used as inputs to our segmentation algorithm. Then for each image, our implemented Matlab code is run for 5 times with various  $N_p$  and performance of the results is calculated. To compute the performance of GOIS approach, we should firstly calculate the value of error using Equation 5-2.

$$\text{Error} = \text{real number of objects} - \text{number of detected objects} \quad (5-2)$$

The error is a quality measurement of the algorithm. Equation 5-3 defines the performance which depends on the error value and number of generations.

$$P(\text{Performance}) = \frac{1}{\text{Error} \times \text{Number of generations}} \quad (5-3)$$

Then performance is high with very small error and a few numbers of generations. Comparison of the performances resulting from different images with different population size is illustrated in Table 5-5.

Table 5-5. Comparison of the Performance Resulting from Different Images with Different Population Size

<b>Population size</b>	<b>Performance of Lena image × 10<sup>5</sup></b>	<b>Performance of Yacht image × 10<sup>5</sup></b>	<b>Performance of Fruit image × 10<sup>5</sup></b>	<b>Average</b>
<b>10</b>	18.4	2.49	5.84	8.91
<b>15</b>	3.4	5.73	7.92	5.68
<b>20</b>	26.88	21.73	13.41	20.67
<b>25</b>	30.0	35.6	19.4	<b>28.34</b>
<b>30</b>	18.5	37.94	20.23	26.0
<b>35</b>	12.748	19.64	11.69	14.69

In Table 5-5, the largest total performance belongs to the generation with 25 individuals.

Computing the total performance of population size= 25 is illustrated in details in Table 5-6.

Table 5-6.Computing the Total Performance of Population Size= 25

Population size	Performance of Lena image		Performance of Yacht image		Performance of Fruit image	
	Iterations	$P \times 10^5$	Iterations	$P \times 10^5$	Iterations	$P \times 10^5$
<b>25</b>	Trial 1	39.0	Trial 1	35.0	Trial 1	30.0
	Trial 2	5.0	Trial 2	57.0	Trial 2	5.0
	Trial 3	40.0	Trial 3	13.0	Trial 3	10.0
	Trial 4	54.0	Trial 4	50.0	Trial 4	33.0
	Trial 5	12.0	Trial 5	13.0	Trial 5	19.0
<b>Average= 28.34</b>	<b>30.0</b>		<b>35.6</b>		<b>19.4</b>	



Comparison of FCM, SGA and GOIS approaches using various images are shown in Table 5-7.

Table 5-7. Comparison of FCM, SGA and GOIS Approaches Using Various Images

<b>Image</b>	<b>Approach</b>	<b>Segmented regions</b>	<b>Computation Time(s)</b>
<b>Lena</b>	FCM	84	69.17
	SGA	47	243.88
	GOIS	18	97.01
<b>Husky</b>	FCM	23	46.27
	SGA	26	52.33
	GOIS	19	50.83
<b>Lake</b>	FCM	103	30.43
	SGA	62	310.39
	GOIS	54	194.38

Through the comparison of these three segmentation approaches in Table 5-7, it can be concluded that the computation time of FCM is less than SGA [1] and GOIS approach. However, the over-segmentation problem is likely to occur. SGA is very

slow because it requires numerous generations to attain an acceptable solution. However, GOIS can obtain the best solution within a reasonable computation time. GOIS is a common purpose algorithm which is capable to split a variety of images. The experimental results expose that GOIS, undoubtedly has the following advantages:

- (1) Continuous contour: as a result of split/merge approach in GOIS, the contours of experiment results are continuous [9]. As shown in Figure 5-12 (m)-(o).
- (2) Non-over segmentation: in GOIS approach, various numbers of similar regions are merged into large ones. Therefore, the number of regions is decreased.
- (3) Short computation time: in comparison with the other evolutionary segmentation algorithm which can gain a reasonable fitness value, GOIS has the advantage of short consuming time.
- (4) Robust: the GOIS approach performs well in the aspect of strength upon the following merits:
  - (a) GOIS can be applied well on the corrupted images by different degree of noise deviation and performs the similar results.
  - (b) In comparison with the other segmentation approaches, GOIS demonstrates that it can generate better results.

## 5.5 Stability of the Algorithm

Stability is a representation of the final fitness value resulting from various run numbers. If fitness value of desired result is fluctuated in a small interval then stability of the algorithm is satisfied.

In order to check the stability, we run our segmentation algorithm 20 times and compare the last fitness value in Figure 5-13.

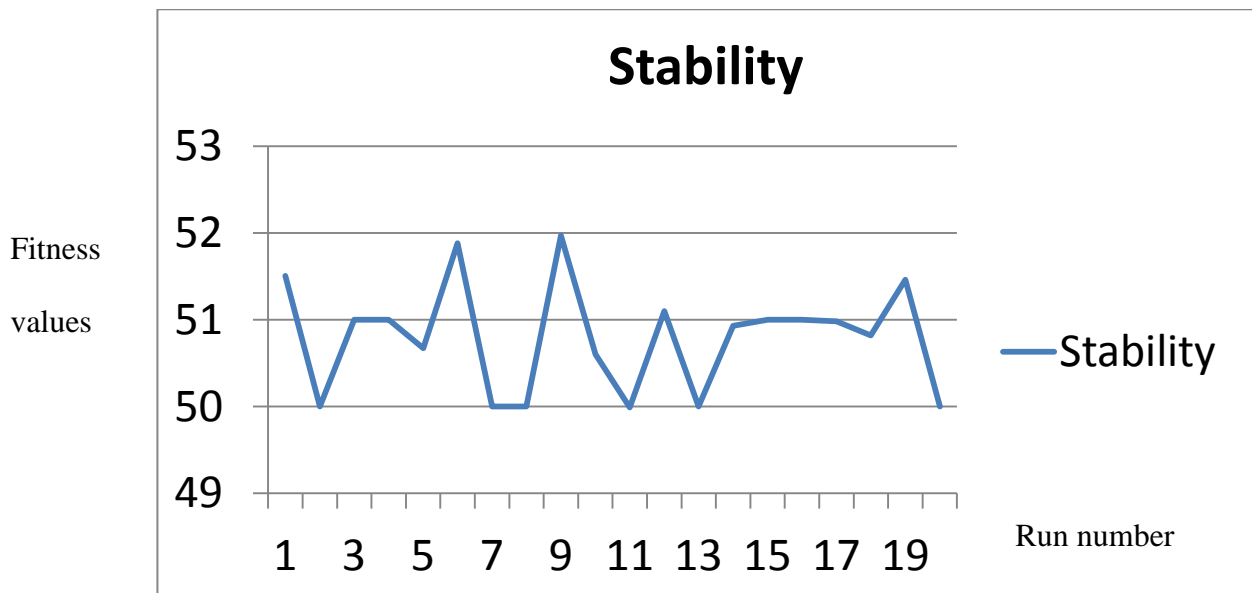


Figure 5-13. Stability of Various Fitness Values

It is concluded from above figure that fitness values are fluctuates between 50 and 52. It means the results of this algorithm are stable.

## Chapter 6

### CONCLUSIONS AND FUTURE WORK

#### 6.1 Conclusions

In this master thesis, we have developed a split/merge image segmentation approach with an evolutionary combinatorial optimization method. This approach for optimized image segmentation employs a K-means clustering algorithm in split process and an intelligent genetic algorithm with efficient chromosome encoding model in merge process. Well-organized chromosome encoding is defined to represent detected regions and a two-point crossover model is applied on chromosomes to produce new generations. GOIS can simultaneously gain the following five objectives: (1) detected edges are always continuous; (2) final detected regions can indicate significant objects from the image; (3) short computation time; (4) this method is practical with acceptable performance for noisy images; (5) high accuracy in comparison with other segmentation methods. Elevated performance of GOIS is shown in terms of assessment performance and computation time compared to some well-known segmentation methods. It is also illustrated that GOIS can be applied on various nature images with different characteristics and produced robust and efficient results.

## **6.2 Future Work**

In order to lower computation time, an effective local search method for computational algorithm, the Tabu search, can be used. Additionally, multi-objective versions of the presented algorithm can be developed to get optimized segmentation according to the multiple objectives.

## REFERENCES

- [1] S. Ying Ho, K. Zheng Lee, "Design and Analysis of an Efficient Evolutionary Image Segmentation Algorithm," *Journal of VLSI Signal Processing*, vol. 35, pp. 29-42, 2003.
- [2] K. S. Fu, J. K. Mei, "A survey on Image Segmentation," *Pattern Recognition*, vol. 13, pp. 3-16, 1981.
- [3] S.M. Bhandarkar, H. Zhang, "Image Segmentation Using Evolutionary Computation," *Transaction on Evolutionary Computation*, vol. 3, no. 1, pp. 1-21, 1999.
- [4] N. R. Hal, S. K. Hal, "A Review on Image Segmentation Techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277-1294, 1993.
- [5] M. Gudmundsson, E. A. El-Kwae, M. R. Kabuka, "Edge Detection in Medical Images Using A Genetic Algorithm," *Medical Images*, vol. 17, no. 3, pp. 469-474, 1998.
- [6] Y. L. Chang, X. Li, "Adaptive Image Region-Growing," *Transaction on Image Processing*, vol. 3, no. 6, pp. 868-872, 1994.
- [7] R. Adams, L. Bischof, "Seeded Region growing," *Transaction on Pattern Analysis and Maching Intelligence*, vol. 16, no. 6, pp. 641-647, 1994.
- [8] S. A. Hojjatoleslami, J. Kittler, "Region Growing: A New Approach," *Transactions on Image Processing*, vol. 7, no. 7, pp. 1079-1084, 1998.
- [9] K. Haris, S. N. Efstratiadis, N. Maglaveras, A. K. Katsaggelos, "Hybrid Image Segmentation Using Watersheds and Fast Region Merging," *Transaction On Image Processing*, vol. 7, no. 12, pp. 1684-1699, 1998.

- [10] J. M. Gauch, "Image Segmentation and Analysis via Multiscale Gradient Watershed Hierarchies," *Transaction On Image Processing*, vol. 8, no. 1, pp. 69-79, 1999.
- [11] A. Tremeau, P. Colantoni, "Regions Adjacency Graph Applied to Color Image Segmentation," *Transaction On Image Processing*, vol. 9, no. 4, pp. 735-744, 2000.
- [12] M. R. Rezaee, P. M. J. Vanderzwet, B. P. F. Lelieveldt, R. j. Vandergeest, J. H. C. Reiber, "A multiresolution Image Segmentation Technique Based on Pyramidal Segmentation and Fuzzy Clustering," *Transaction On Image Processing*, vol. 9, no. 7, pp. 1238-1248, 2000.
- [13] D. N. Chun, H. S. Yang, "Robust Image Segmentation Using Genetic Algorithm with A Fuzzy Measure," *Pattern Recognition*, vol. 29, no. 7, pp. 1195-1211, 1996.
- [14] T. N. Pappas, "An Adaptive Clustering Algorithm for Image Segmentation," *Transaction On signal Processing*, vol. 40, no. 4, pp. 901-914, 1992.
- [15] T. Pavlidis, "Algorithm for Graphics An Image Processing," in *MD: Computer Science*, Rockville, 1982.
- [16] Y. A. Tolias, S. M. Panas, "Image Segmentation by A Fuzzy Clustering Algorithm Using Adaptive Spatially Constrained Membership Functions," *Transaction on Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 359-369, 1998.
- [17] Bezdek, J. C., "Pattern Recognition with Fuzzy Objective Function Algorithm," in *Plenum press*, newyork, 1981.
- [18] S. Kirkpatrick, C. Gelatt, Jr., M. Vecchi, "Optimization by Simulated

- Annealing," vol. 220, no. 4598, pp. 498-516, May 1983.
- [19] B. A. Berg, "Locating Global Minima in Optimization problems by A Random-Cost Approach," vol. 361, pp. 708-710, 1993.
- [20] Y. Wang, R. A. Prade, J. Griffith, W.E. Timberlake, J. Arnold, "A Fast Random Cost Algorithm for Physical Mapping," vol. 91, pp. 11094-11098, 1994.
- [21] M. Creutz, "Microcanonical Monte Carlo Simulation," vol. 50, no. 19, pp. 1411-1414, 1983.
- [22] S. W. Mahfoud, D. E. Goldberg, "Parallel Recombinative Simulated Annealing: A Genetic Algorithm," *Parallel Comput*, vol. 21, no. 1, pp. 1-28, 1995.
- [23] M. B. Salah, I. B. Ayed, A. Mitiche, "Multiregion Graph Cut Image Segmentation," in *Computer Vision Theory and Applications*, 2008.
- [24] Y. Feng, Z. Wang, X. N. Wang, "Ant Colony Optimization for Image Segmentation," in *Machine Learning and Cybernetics*, 2005.
- [25] D. Liu, Y. Xiong, K. Pulli, L. Shapiro, "Estimating Image Segmentation Difficulty".
- [26] R. C. Gonzalez, R. E. Woods, "Real-time Digital Image Enhancement," *Digital Image Processing*, vol. 69, no. 5, 1981.
- [27] Canny, J. F., "A computational Approach to Edge detection," *Pattern Analysis and machine Intelligence*, vol. 8, pp. 679-698, 1986.
- [28] J. G. Park, C. Lee, "Skull Stripping Based on Region Growing for Magnetic Resonance Brain Images," vol. 47, no. 4, pp. 1394-1407, 2009.
- [29] A. Albiol, L. Torres, E.J. Delp, "An Unsupervised Color Image Segmentation Algorithm for Face Detection Applications," in *Image Processing*, 2001.



- [30] J. H. Holland, *Adaptation in Natural and Artificial Systems.*, M. P. B. Books, Ed., London, 1998.
- [31] D. Goldberg, *Genetic Algorithm in Search*, MA: Addition-Wesley, 1989.
- [32] M. Mitchell, *An Introduction to Genetic Algorithms*, MA: MIT Press, 1996.
- [33] Y. J. Zhang, "A Survey on Evolutionary Methods for Image Segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335-1346, 1996.
- [34] A. Caprara, H. Kellerer, U. Pferschy, D. Pisinger, "Approximation Algorithms for knapsack Problems with Cardinality Constraints," vol. 123, no. 8, pp. 333-345, 2000.
- [35] S. M. Bhandarkar, Y. Zhang, W. D. Potter, "An Edge Detection Technique Using Genetic Algorithm-based Optimization," *Pattern Recognition*, vol. 27, no. 9, pp. 1159-1180, 1994.
- [36] L. Davis, *Handbook of Genetic Algorithm*, L. Davis, Ed., New York, 1991.
- [37] V. Gleb, Tcheslavski, "Image Restoration and Reconstruction: Reconstruction Preliminaries Basic Techniques and Additive Noise Only," 2008.
- [38] C. W. Chen, J. Luo, K. J. Parker, "Image Segmentation via Adaptive K-means Clustering and Knowledge-based Morphological Operations with Biomedical Applications," *Image Processing*, vol. 7, no. 12, pp. 1673-1683, 1998.
- [39] Y. J. Zhang, "A Survey on Evolutionary Methods for Image Segmentation," *Pattern Recognition*, vol. 29, no. 8, pp. 1335-1345, 1996.
- [40] T. V. Mathew, "Genetic Algorithm," Indian Institute of Technology Bombay, Mumbai.
- [41] H. Zhang, "Image Segmentation Using Combinatorial Optimization, master

thesis," 1997.