MANAGING GENETIC ALGORITHM PARAMETERS TO IMPROVE SEGGEN A THEMATIC SEGMENTATION ALGORITHM

(GENETİK ALGORİTMA PARAMETRELERİNİ KULLANARAK SEGGEN TEMATİK SEGMENTASYON ALGORİTMASININ GELİŞTİRİLMESİ)

by

Neslihan Şirin Saygılı, B.S.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

INSTITUTE OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

July 2013

MANAGING GENETIC ALGORITHM PARAMETERS TO IMPROVE SEGGEN A THEMATIC SEGMENTATION ALGORITHM

(GENETİK ALGORİTMA PARAMETRELERİNİ KULLANARAK SEGGEN TEMATİK SEGMENTASYON ALGORİTMASININ GELİŞTİRİLMESİ)

by

Neslihan Şirin Saygılı, B.S.

Thesis

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE

Date of Submission : July 23, 2013
Date of Defense Examination: July 26, 2013

Supervisors : Prof. Dr. Bernard Levrat Assoc.
Prof. Dr. Tankut Acarman
Committee Members : Assoc. Prof. Dr. Temel Öncan
Asst. Prof. Dr. Müjde Erol Genevois
Dr. Vincent Labatut

ACKNOWLEDGEMENTS

I would like to present my sincere gratitude to Prof. Dr. Bernard Levrat and Assoc. Prof. Dr. Tankut Acarman, whose expertise, understanding, and patience, added considerably to my graduate experience.

I would also like to thank my family for the support they provided me through my entire life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Due to the remarkable increase in the number of available text databases in last decades, the need for efficient searching methods has become a major challenge for information retrieval. Moreover, efficiency in the accessibility to the relevant information, satisfying the information needs of the user is now becoming a crucial issue in the choice of a searching system. Since document structure of a document search base is not specifically built for the task, and this is particularly the case for the web, users generally consider only small parts of documents returned as response to their queries as being relevant. This is one of the reasons thematic text segmentation has taken more importance where the aim is to access directly to the parts of the documents containing the relevant information more efficiently than in traditional information retrieval where only whole texts are considered. Thematic segmentation can be defined as the process of separating written texts into meaningful homogeneous units in accordance with the criteria stated in Salton et al.'s (1996) definition which states that thematic segmentation of a text is its splitting into segments such that the internal cohesion of segments and the dissimilarity between adjacent segments is maximum. SegGen (Lamprier et al., 2007) is a linear thematic segmentation algorithm grounded on a variant of the Strength Pareto Evolutionary Algorithm (Zitzler, 1999) and aims at optimizing the two criteria of the Salton's definition of segments: a segment is a part of text whose internal cohesion and dissimilarity with its adjacent segments are maximal. This thesis describes improvements that have been implemented in the approach taken by SegGen by tuning the genetic algorithm parameters according with the evolution of the quality of the generated populations. Two kinds of reasons originate the tuning of the parameters and have been implemented here. The first one rests on general principles of *autonomous search* (Hamadi et al., 2008), which consists in modifying the parameters and operators of the genetic algorithm along with the increasing quality of the generated population through the generations. The second type of improvements is also to take into account the increasing of quality of the

population as the process evolves, but to do so by taking into account the nature of the coding of individuals which in this case are segmentation instances represented by binary vectors corresponding to the positions of the boundaries of the segmentations.

Keywords: Thematic Segmentation, Genetic Algorithm, Multi-Objective Optimization Problem.

RESUME

En raison de l'accroissement considérable du volume de données textuelles accessibles en particulier au travers du Web lors de ces dernières années, le besoin de méthodes de recherche efficaces est devenu un défit majeur de la recherche d'informations.

Par ailleurs l'efficacité de l'accès de l'utilisateur à l'information pertinente comme réponse à une requête est un critère déterminant du choix d'un système de recherche d'informations. En raison du fait que la structure des documents d'une base documentaire ne s'appuie généralement pas sur le type d'utilisation qui en est faite, et c'est le cas pour les documents interrogés sur le Web, les utilisateurs considèrent en général que seule une très petite partie de chaque document fourni par un système en réponse à une question est susceptible d'être véritablement pertinente. Ceci constitue l'une des raisons qui explique l'importance donnée ces dernières années à la segmentation thématique automatique des textes pour des utilisations de type recherche de passages (en anglais, *passage retrieval*) dans lesquelles les réponses du système à une requête sont constituées de parties pertinentes de documents et non de documents pertinents fourni dans leur entier.

La segmentation thématique peut être définie comme une tâche visant à séparer un texte écrit, en unité significatives homogène conformément à la définition de Salton et al. (1996) qui la présente comme le découpage d'un texte en un ensemble de segments adjacents tels que, du point de vue sémantique, la cohésion interne des segments et la dissimilarité entre des segments adjacents soient maximales.

Parmi les algorithmes de segmentation existant SegGen (Lamprier et al., 2007) est l'un de ceux qui obtient les meilleures performances. Il s'appuie sur la définition de Salton

énoncée précédemment et pose le problème de segmentation comme un problème d'optimisation bicritère ou les deux critères en jeu sont ceux de cette définition.

Pour le résoudre, il utilise alors une variante de l'algorithme SPEA (Strength Pareto Evolutionary Algorithm (Zitzler, 1999). Alors que SegGen utilise une version générique de l'algorithme nous montrons dans cette thèse comment en tenant compte de l'état d'avancement de la solution et en adaptant le codage aux données spécifiques sur lesquelles porte l'algorithme les résultats de SegGen peuvent être très sensiblement améliorés.

Mots clés : Segmentation Thématique, Algorithme Génétique, Problème d'optimisation multi-objectif.

ÖZET

Son yıllarda kullanılabilir metin veritabanı sayısında ciddi bir artış meydana gelmiştir. Buna bağlı olarak, bilgi edinme alanında etkili araştırma yöntemlerine duyulan ihtiyaç da artmıştır. Ayrıca ilgili bilgiye erişimde etkinlik ve kullanıcıların bilgi ihtiyacına doyurucu yanıtlar verilmesi günümüzde arama sistemi seçiminde önem kazanmaktadır. Belge arama temelinin belge yapısı, hizmet ettiği görevlere özel olarak tasarlanmadığından ve bu durum web ortamı için geçerli olduğundan genellikle kullanıcılar sorgularının yanıtı olarak gönderilen belgelerin ancak az bir bölümünün aramayla ilgili olduğunu düşünmektedir. Bu nedenle parça kurtarma olarak bilinen dönen belge parçalarına erişebilmek amacıyla çeşitli araştırmalar yapılmıştır (Hearst & Plaunt, 1993; Callan, 1994). Dijital kütüphanelerde otomatik metin özetleme kullanımı potansiyel faydalar getirse de bu durum, özet oluşturmada kullanılan etkin metin bölümleme gibi araçların yardımına bağlıdır. McDonald ve Chen (2002), bölümleme yaklaşımının bir özet belgesinde geçen konular hakkında yüksek temsil kabiliyetine sahip olduğunu belirtir. İşte bu nedenledir ki metinsel belgelerin tematik bölümlenmesi bu alanda daha fazla önem kazanmış olup metinden anlamsal bütünlük içeren bölümleri edinmenin önünü açmıştır. Bu işlemin amacı, doğrudan ilgili bilgiyi ihtiva eden belge parçalarına geleneksel bilgi kurtarma yöntemlerine göre daha etkili erişim sağlayabilmektir. Çünkü geleneksel yöntemler metni bir bütün olarak alır.

Anlamsal bölümleme, yazılı metni Salton'un (et al., 1996) yaptığı tanımda geçen kriterlere göre anlamlı homojen parçacıklara ayırma süreci olarak tanımlanabilir. Bahsi geçen tanıma göre bir metnin anlamsal olarak bölümlenmesi onu parçalara yani segmentlere ayırmak demektir. Bu işlemde bölümlerin iç bütünlüğü ve birbirine komşu bölümler arasındaki farklılıklar hat safhadadır. Bu tanıma göre otomatik metin bölümlenmesi, bu kriterlere uygun bir belge içerisinde sınırları belirleyerek belli başlı tematik ayrımların tayin edilmesi şeklinde anlaşılabilir. SegGen (Lamprier et al., 2007)

ise Strength Pareto Evolutionary Algoritmasının bir varyantı üzerine şekillendirilmiş bir anlamsal bölümleme algoritmasıdır. Bu algoritma ile hedeflenen, Salton'un bölümleme tanımına dair iki kriterin optimize edilmesidir. Kriterler ise bir bölümün ait olduğu metnin kendi içinde maksimum bütünlüğe sahip olması ve komşu bölümlerle arasında minimum benzerlik olması şeklindedir.

Bu tez çalışmasında, elde edilen popülasyonların niteliğinin evrimine göre genetik algoritma parametrelerinin ayarlanması suretiyle SegGen yaklaşımı üzerinde uygulanan birtakım gelişmeler anlatılmaktadır. Parametre ayarları iki farklı nedene dayandırılmış ve bu tez çalışması kapsamında uygulanmıştır. Birinci nedene göre; popülasyonun niteliğine ilişkin genel kriterlere göre değerlendirme yapılabileceğinden elde edilen popülasyonların genel niteliği, süreç ilerledikçe artar ve parametrelere değer koymak ve arama sürecinde gücü artırırken çeşitlilik faktörlerini azaltan yeni operatörler tanımlamak mantıklı görünmeye başlar. Diğer nedene göre ise; popülasyonlar içerisindeki öğeler makul metin bölümleri olduğundan mevcut bölümler içerisindeki cümlelere, optimizasyonu söz konusu iki kriterde gömülü cümleler arasındaki benzerliklerin analizi açısından, bağlı bulundukları sınırlara olan uzaklıklarına göre değer yüklemek gerekir.

Anahtar Sözcükler: Anlamsal Bölümleme, Genetik Algoritma, Çok Amaçlı Eniyileme Problemi.

1 INTRODUCTION

Due to the huge increase in the number of available text databases in recent years, the need for efficient searching methods has become a major challenge for information retrieval. Moreover efficiency in the accessibility to the relevant information, satisfying user's information need is now becoming a crucial issue in the choice of a searching system. Because documents of a document search base are not specifically built for the task they are used for, and this is particularly the case for the web, users generally consider only small parts of documents returned as response to their queries as being relevant. This is one of the reasons for which researches were initiated in the aim to give access to parts of returned documents and constitutes a subfield of the domain of information retrieval known as passage retrieval (Hearst & Plaunt, 1993; Callan, 1994).

Thematic segmentation can be characterized as the process of separating written text into meaningful homogeneous units by determining the positions at which topics change in a stream of text. Using automatic text summarization in digital libraries offers potential benefits but this is dependent on having tools like efficient text segmenter built the abstracts. McDonald and Chen (2002), for example, remark that segmentation is a good way to thoroughly ensure the representation of the various topics of a document in a summary. In information retrieval, the burden to retrieve information relevant to a query in large texts is a drawback due in particular to the fact that the documents have not specifically been conceived to answer to the particular query for which they are furnished as a response. This is another reason why the thematic text segmentation of textual documents has taken more importance in this domain and has given rise to passage retrieval, as a subfield of information retrieval, where the aim is to access directly to the parts of the documents containing the relevant

information more efficiently than in traditional information retrieval where only whole texts are considered (Salton et al., 1996).

SegGen (Lamprier et al., 2007) is a linear thematic segmentation algorithm grounded on the definition of thematic segmentation of a text given by Salton (et al., 1996) according to whom thematic segmentation of a text is its splitting into parts where the internal cohesion and the dissimilarity between adjacent segments are maximal. According to this definition, the main aim of SegGen is to find out the boundaries between subtopics, such that in the resulting segments, internal coherence and dissimilarity between adjacent ones are maximal. To achieve this, SegGen states this problem as an optimization problem aiming at maximizing these two last criteria and uses a multi-objective genetic algorithm for this task. Unlike other classical segmentation methods where boundaries are sequentially put one after the other, SegGen algorithm permits to have a global view on all the potential segments to take a decision since all the boundaries between potential segments are set at the same time.

This thesis describes improvements that have been implemented in the approach taken by SegGen by tuning the genetic algorithm parameters depending on the evolution of the quality of the generated populations.

1.1 Purpose of the Thesis

One of the main drawbacks to the majority of existing segmentation methods, is that the criteria used to set boundaries between segments are local in the sense that the relationships or similarities between sentences are examined locally nearby the potential segments under consideration and do not take the whole potential segmentation into account. Roughly sketched in such methods, thematic similarities between segments are calculated on the basis of the distribution of the meaningful lexical inventory in each segment. And for that, a lot of the existing segmentation methods use a sliding window to find out dissimilarity measures in consecutive positions of the sliding window or values of some cohesion. While calculating the thematic similarity, it is considered whether sentences are in windows. Moreover, the efficiency of such methods is very dependent on the dimension of the size of the sliding

windows. Lamprier et al. (2008) indicates that small modifications of the window size could greatly influence setting of the boundaries between segments leading to over or under segmentation of the text depending on a too small or too large window size.

Contrary to these algorithms which rest on sliding windows and set the boundaries between segments on local criteria, SegGen algorithm proposes an original and efficient way to cope with the problem of linear text segmentation that allows having a global view on all the potential segments to take a decision since all the boundaries between potential segments are set simultaneously.

The main purpose of this thesis is to understand research, work on thematic segmentation algorithms, and to improve SegGen algorithm. Different thematic segmentation approaches existing in literature and SegGen algorithm are examined. Then, several improvements to SegGen algorithm are proposed, implemented and their performance are analyzed.

Proposed several improvements of SegGen algorithm are guided by two main ideas, which inspired autonomous search (Hamadi et al., 2008). The first one rests on general principles of *autonomous search*, which consists in modifying the parameters and operators of the genetic algorithm along with the increasing quality of the generated population through the generations. The second type of improvements is also to take into account the increasing of quality of the population as the process evolves, but to do so by taking into account the nature of the coding of individuals which in this case are segmentation instances represented by binary vectors corresponding to the positions of the boundaries of the segmentations.

1.2 Organization of the Thesis

Chapter 2 presents linear text segmentation and its basic functionalities in its subchapter 2.1, and then chapter 2.2 covers SegGen algorithm and its detailed explanations.

Proposed improvement approach to SegGen algorithm is presented in Chapter 3 in detail, including mutation operator tuning, mutation probability tuning, crossover operator tuning, and fitness function tuning.
Experimental results of proposed method are given in Chapter 4 detailing solution retrieval, experimental process, and result comparison.

Chapter 5 is the review of the thesis and the conclusion.

2 BASICS OF LINEAR TEXT SEGMENTATION AND SEGGEN

2.1 Linear Text Segmentation and its Basic Functionalities

The first part of this section is devoted to the basics of information retrieval, and its subfields such as in particular the core of search engines and text segmentation. Text segmentation implies that a document may contain multiple topics, and the task of computerized text segmentation may be to discover these topics automatically and segment the text accordingly. The topic boundaries may be apparent from section titles and paragraphs. In other cases the approach needs to use techniques similar to those used in document classification. Segmenting the text into topics can be useful in some natural language processing tasks: it can improve information retrieval efficiency significantly by indexing/recognizing documents more precisely or by giving the specific part of a document corresponding to the query as a result. Tokenization, clearing stop-words, case folding and stemming are pretreatments before the thematic segmentation of the document (Manning et al., 2008). An attempt to measure the degree to which a document matches a query or the score of a document for a query prompts the development of pretreatments such as term weighting and the computation of scores.

2.1.1 The Pretreatments Before Thematic Segmentation

Tokenization: Tokenization is the process of chopping strings into meaningful pieces. A token is an instance of a useful semantic sequence of characters in text as seen in Figure 2.1.

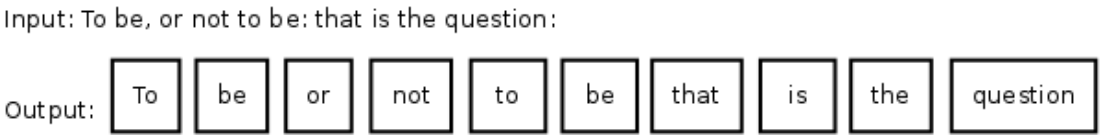


Figure 2.1 An example of tokenization.

Dropping common terms: Some words are used extensively in text, and therefore the contribution of text to keep search is smaller than meaningful words. These words are called stop words, which are filtered out before the main process. Following some examples of stop words:

"*a, and, any, as, by, etc, for, in, kg, my, of, per, to*"

Case folding: Since either uppercase or lowercase of a word usually has the same value in the text, the general approach is to do case folding by reducing all letters to lowercase.

"*Prime Minister Benazir Bhutto will visit China on Feb. 11*"
should become
"*prime minister benazir bhutto will visit china on feb. 11*"

Stemming: The aims of stemming are to reduce derivative forms into base forms by chopping off the derivational affixes; documents use different forms of a word. A simple example of stemming is below,

toy, toys, toys', toy's -> toy
fishing, fished, fish, fisher -> fish

Porter Stemmer algorithm (Porter, 1980) is one of the most widespread known stemming algorithms for English language. Figure 2.2 shows a paragraph of sample text and its Porter Stemmer algorithm output (Woolf, 2003).

Sample text: A woman knows very well that, though a wit sends her his poems, praises her judgment, solicits her criticism, and drinks her tea, this by no means signifies that he respects her opinions, admires her understanding, or will refuse, thought the rapier is denied him, to run through the body with his pen.

Porter Stemmer output: A woman know very well though a wit send her hi poem prais her judgment solicit her critic and drink her tea thi by no mean signify that he respect her opinion admir her understand or will refus thought the repier is deni him to run through the bodi with hi pen

Figure 2.2 An output of Porter Stemming algorithm on sample text.

2.1.2 Document Representation

Tokenization, clearing stop-words, case folding and stemming are pretreatments before the thematic segmentation of the document. On an ongoing basis, vector representation of documents, term frequency, inverse document frequency, and cosine similarity measure concepts are used to clarify great importance of representation of basic units and number of occurrence of basic units in the document (Manning et al., 2008). There are many types of document representation and similarity metrics; we used vector representation and cosine similarity in this thesis.

Vector space model: Vector space model can be defined as an algebraic model of the representation of a set of text documents as vectors. Each term t of the dictionary is considered as a dimension and a document can be represented by the weight of each dictionary term as following:

$$\overrightarrow{V}(d) = (w(t_1, d), w(t_2, d), \ldots., w(t_n, d)) \quad (2.1)$$

Vector representation does not take into account ordering of words in a document. For example, “Alice is quicker than Bob” and “Bob is quicker than Alice” have the same vectors. As shown in Figure 2.3 (Manning et al., 2008), there are three-vector representations of documents. Three sample words and corresponding weight of term values.

Dictionary	V(d1)	V(d2)	V(d3)
affection	0.996	0.993	0.847
jealous	0.087	0.120	0.466
gossip	0.017	0	0.254

Figure 2.3 An example of vector representation

Term frequency: The number of times a term (t) occurs in a document (d) is called term frequency. It is shown in the form of $tf_{t,d}$.

Document frequency: Document term frequency can be defined as the number of documents in the document collection that contain a term t. It is denoted as df_t.

Inverse document frequency: The inverse document frequency of a term is a measure of general importance of the term. Inverse document frequency is the number of documents in document collection that comprise term t. N represents the total number of the document base; the inverse document frequency of a term t is formulated as follows:

$$idf_t = \log \frac{N}{df_t} \tag{2.2}$$

If we use only term frequency, it causes a critical problem that all terms are considered equally important. Since some terms have little or no discriminating power in determining relevance; inverse term frequency renders spreading of terms in the document base. For example, a document collection on the auto industry has the term auto in almost every document. So we should use a mechanism for reducing the effect of terms that occur often in the collection to be meaningful for relevance determination.

Term t	dft	idft
car	18165	1.65
auto	6723	2.08
insurance	19241	1.62
best	25235	1.5

Figure 2.4 An example of term frequency and inverse document frequency

As shown in Figure 2.4 (Manning et al., 2008), there are term frequency values and inverse document frequency values of some terms from 806,791 documents. If the term has a great number of df_t values, it has lower number of idf_t value. Inverse document frequency permits to reduce a great number of df_t values. From this point, we now combine the definitions of term frequency and inverse document frequency, to produce a composite weight for each term in each document that can be defined as following:

$$w(t,d) = tf - idf_{t,d} = tf_{t,d}\ x\ idf_t \tag{2.3}$$

If the weight of a term is:

- High, t occurs many times in a small set of documents
- Low, t occurs fewer times in a document, or t occurs in many documents
- Very low, t occurs in almost every document

Similarity metrics: Similarity between sentences or paragraphs can be represented in various ways such as Jaccard similarity coefficient and cosine similarity. First, the Jaccard coefficient measures similarity between given word sets, and can be defined as the size of the intersection divided by the size of the union of the sample word sets.

$$jaccard(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{2.4}$$

The second similarity metric is cosine similarity method. In this similarity metric, word is used as a vector to find the normalized dot product of the two documents. By

using the word frequencies for each document, the normalized dot product of the frequencies can be used as a measure of similarity. For cosine similarities resulting in a value of 0, the angle between the objects is 90 degrees because the documents do not share any words. The general approach of evaluating the similarity between two documents d1 and d2 is to assess the cosine similarity of vector representations of documents $\vec{V}(d_1)$ and $\vec{V}(d_2)$.

$$sim(d_1, d_2) = \frac{\vec{V}(d_1)\cdot\vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d2)|} \tag{2.5}$$

Cosine similarity is a measure of similarity between two vectors by an inner product that measures the angle between given two document vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$ (Manning and Schutze, 1999). Cosine measure is the cosine of the angle θ between the two vectors $\vec{V}(d_1)$ and $\vec{V}(d_2)$, shown in Figure 2.5 (Slidewiki, 2013).

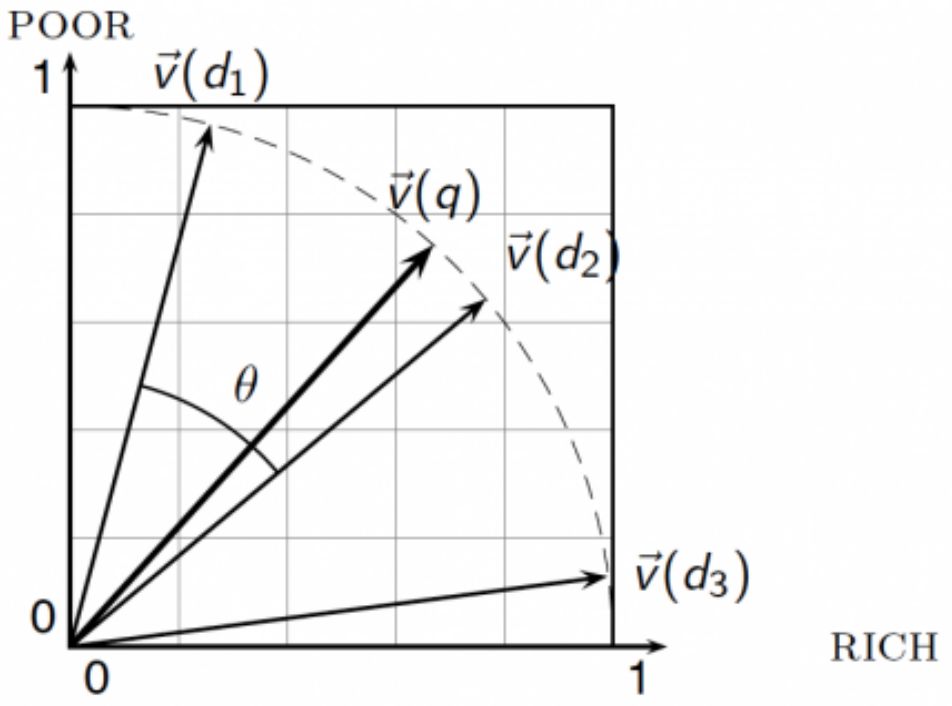


Figure 2.5 Cosine similarity illustration.

2.1.3 Segmentation Process

Following the preliminary steps, text segmentation could be seen as determining the most important thematic breaks by setting the boundaries in a document grounded on previous calculation. Besides methods grounded on linguistics marks (Mochizuki et al., 1998), there are many segmentation methods that rely on statistical approaches, such as TextTiling (Hearst, 1997), C99 (Choi, 2000), DotPlotting (Reynar, 2000), Segmenter (Kan et al., 1998). Many existing segmentation methods used to determine boundaries between segments are local boundaries. In other words, similarities between sentences are checked locally nearby the potential segments and do not take into account the whole potential segmentation. In such methods, thematic similarities between are calculated on the basis of the distribution of the meaningful lexical inventory. The common point of statistical segmentation methods is that they determine the thematic changes via lexical inventory variations; so, for example, they set the boundaries by using sliding windows on the text to measure the variation of the level of local cohesion, setting the boundaries where local cohesion is the lowest. A lot of existing segmentation methods define a sliding window to find out dissimilarity measures in consecutive positions of the sliding window. For instance; TextTiling (Hearst, 1997) algorithm uses a sliding window, which determines *block*s in the text, and calculates the value of the dissimilarity of adjacent blocks based on differences between lexical inventories in adjacent blocks (in fact, it uses a vectorial representation of textual units and the measure cosine for that). Thematic changes are detected on the basis of the evolution of the dissimilarities between adjacent sliding blocks. Thus, significant vocabulary changes are seen at points with subtopic change. On the other hand, there is a different algorithm from previously mentioned, ClassStruggle (Lamprier et al., 2007) based on an initial clustering of the sentences of the text. ClassStruggle uses an initial clustering of the sentences of the text based on their similarity, in order for a global view on their semantic relations. The resulting clusters evolve by considering their proximity in the text. This preliminary partitioning provides a global view on the sentences relations existing in the text, taking into consideration the similarities in a group rather than individually. ClassStruggle is based on the distribution of the occurrences of the members of each class.

2.2 SegGen Algorithm

SegGen (Lamprier et al., 2007) is a linear thematic segmentation algorithm grounded on the definition of thematic segmentation of a text given by Salton (et al., 1996) according to whom thematic segmentation of a text is its splitting into parts where the internal cohesion and the dissimilarity between adjacent segments are maximal. With reference to this definition, the main aim of SegGen is to find out the boundaries between subtopics, such that in the resulting segments, internal coherence and dissimilarity between adjacent ones are maximal. To achieve this SegGen states this problem as an optimization problem aiming at maximizing these two last criteria and uses a multi-objective genetic algorithm for this task. In SegGen, the main aim is to find out the subtopics, which create internal coherence and are distinguished from other parts of the text. Hence, the algorithm has two objective functions such as internal cohesion and dissimilarity between adjacent parts. SegGen uses a variation of Strength Pareto Evolutionary Algorithm (Zitzler, 1999); so it can be classified as an elitist evolutionary multi-objective algorithm. (Elitism can be described as retaining the best individuals in a generation unchanged in the next generation.) Following section, firstly genetic algorithm is examined; after that SegGen is explained in more detail.

2.2.1 A General Explanation of Genetic Algorithms

Genetic algorithms (GAs) were devised by Holland and then developed by Holland, his students and colleagues in the 1960s and the 1970s. Although, at the beginning the aim of the study was not to propose algorithms to solve specific problems, they ended up with develop the mechanisms of natural adaptation could be imported into computer systems. Adaptation in Natural and Artificial Systems (Holland, 1975) introduced genetic algorithms as an abstraction of biological evolution. After this historical background, genetic algorithms can be defined as that use methods based on the process of natural evolution. Genetic algorithms can be classified in evolutionary algorithms (EA) that generate solutions to optimization problems using techniques inspired by natural evolution.

The simplest representation of individual for genetic algorithms is a string of bits is known as a ***chromosome***, and each bit is known as a ***gene***. A chromosome is usually taken to represent an entire ***individual*** within the population. An individual could coding depends on problem such as a vector of numbers or a string of letters. The ***population*** consists of a set of individuals. A more general form of genetic algorithm contains three types of operators: selection, crossover, and mutation (Melanie, 1999).

Genetic algorithms use fitness function as a quality measure. ***Fitness function*** is the objective function of the genetic algorithms that evaluates qualification of given individual as a solution to problem by analyzing its genetic content value. As a result the fitness function process, it assigns a fitness value to the given individual.

Selection: This operator picks out individuals in the population for reproduction. Selection operator can be in progress different ways such as fitness proportionate selection or tournament selection. Fitness proportionate selection also known as *roulette wheel selection* that an imaginary proportion of the wheel is assigned to each of the chromosomes based on their fitness value as shown in Figure 2.6 (Wikipedia, 2013). The fitter chromosome has more chance to select than worse one. Roulette wheel selection is a kind of elitist selection that retaining the best individuals in a generation unchanged in the next generation. Another well-known selection method is tournament selection that involves n times roulette selection, which indicates tournament to produce a subset of individuals. The winner of each tournament is in this subset as the selected individuals.

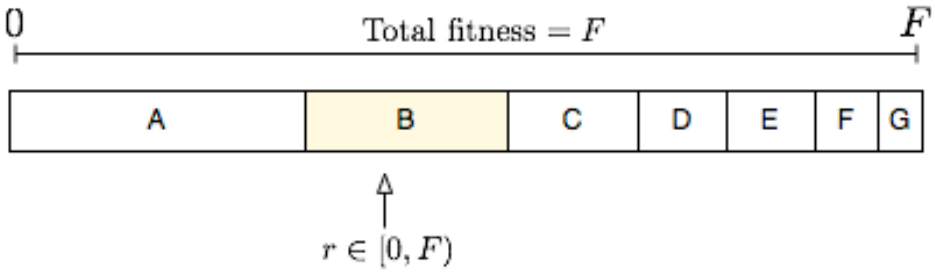


Figure 2.6 Fitness proportionate selection example.

Crossover: This operator randomly chooses a position in two given individual, called parents, and interchanges the subsequences before and after that position between parents to create two offspring. In Figure 2.7 Illustrated an example (single point) crossover operator (Wikipedia, 2013).

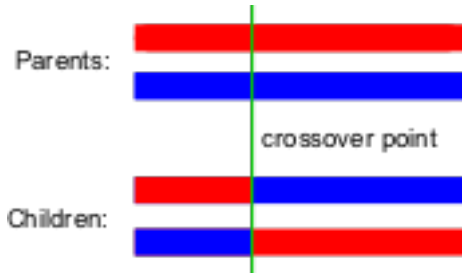


Figure 2.7 Single point crossover operator example.

The crossover operator in GA could be more than one point crossover that allows select multi points to exchange subsequences between given a pair of parent.

Mutation: This operator makes various changes on selected bits in a chromosome. The change depends on probability. The aim of the mutation operator is to sustain genetic diversity from current generation to next generation. Mutation changes one or more gene values or position in a chromosome from its initial state. Owing to the nature of mutation by which the solution may change entirely from the previous solution, GA can reach target solution by using mutation. For different chromosome types, different mutation types are favorable as following,

- Flip bit mutation: The most known mutation type takes the chosen gene and inverts the bits. For example, 1100**1**0110**0** -> 1100**0**0110**1**.
- Order changing: The two bits are selected and exchanged. For example, 123**4**567**8**9 -> 123**7**56**4**89.
- Add or subtract a small value: A small real value is added or subtracted to selected genes. For example, [2.91, 2.11, **5.86**, **4.18**] -> [2.91, 2.11, **5.66**, **4.38**].

General flow of GA: GA working schema is follows:

1. Generate a random population of individuals (this is the first generation).

2. Calculate the fitness value of each individual in the population. If the termination criteria are satisfied, stop. Otherwise, continue with step 3.
3. Repeat the following steps until n offspring have been created:
    a. Pick out a pair of parent from the current population,
    b. With a probability crossover occurs, in the case, which no crossover happens, selects a copy of a couple of parent in population.
    c. With a mutation probability, mutation takes place, and put down the produced individuals in the new population.
4. Replace the current population with the new population.
5. Go to step 2

Each iteration of above process is called a generation. At the end of entire iterations there are often one or more potential goal chromosomes in the population. Due to the major role of randomness in general flow of GA, each execution of GA could make different detailed behaviors. As a result of this point, it is possible to obtain the result as the average of different executions of the program on the same problem.

2.2.2 A Detailed Case of SegGen

SegGen method uses genetic algorithm for text segmentation. SegGen propose an original and efficient way to cope with the problem of linear text segmentation since it states the segmentation problem as a bi-objective optimization problem grounded on the criteria of the Salton's definition of segments previously evocated Unlike other classical segmentation methods where boundaries are sequentially put one after the other, SegGen has a global point of view on the segmentation due to the fact that it sets all the boundaries between segments simultaneously. This global view seems to be more realistic in particular since setting a boundary anywhere should have necessary some effect on other boundaries on not only on contiguous one. Considering the segmentation problem as a bi-objective optimization problem, SegGen is a kind of evolutionary algorithm that evaluates the segmentations of the whole text in preference to setting boundaries incrementally. The lack of knowledge about the structure of the text or in the number of segments to create causes a large search space and leads us to

consider a genetic algorithm to defeat the complexity. To solve the bi-objective optimization problem, SegGen uses an implementation of the multi-objective algorithm SPEA (Zitzler, 1999), a classical multi-objective algorithm. Multiobjective optimization problems are common that involve more than one objective function to be optimized simultaneously. For example the aim of a multi-objective method SegGen: maximization of the internal cohesion of the formed segments and minimization of the similarity of the adjacent segments. Multiobjective optimization problems can be formally defined as follows.

Multiobjective optimization: A general multiobjective optimization problem comprises a set of n decision variables, a set of k objective functions, and a set of m constraints. Objective functions and constraints are functions of the decision variables. The optimization goal is to

$$\begin{aligned} &\textit{maximize} && y = f(x) = \left(f_{1(x)}, f_{2(x)}, \dots, f_{k(x)}\right) \\ &\textit{subject to} && e(x) = \left(e_{1(x)}, e_{2(x)}, \dots, e_{m(x)}\right) \leq 0 \; x = (x_1, x_2, \dots, x_m) \in X \\ &\textit{where} && y = (y_1, y_2, \dots, y_k) \in Y \end{aligned} \tag{2.6}$$

x is the decision vector, y is the objective vector, X is stands for as the decision space, and Y is called the objective space. There is a need for redefinition of the concept of optimality for the case of multiple objectives: Firstly, Pareto dominance is introduced as follows.

Pareto dominance: For any two decision vectors a and b,

$$\begin{gathered} a \succ b \; (a \; dominates \; b) \; iff \; f(a) > f(b) \\ a \succeq b \; (a \; weakly \; dominates \; b) \; iff \; f(a) \geq f(b) \\ a \sim b \; (a \; is \; indifferent \; to \; b) \; iff \; f(a) \not\geq f(b) \wedge f(b) \not\geq f(a) \end{gathered} \tag{2.7}$$

Regarding to the concept of Pareto Dominance, if a is optimal, it cannot improve in any objective without trade-off in both objectives.

Pareto optimality: Such solutions are stand for as Pareto optimal that none of these can be identified as better than the others on given objective functions.
A decision vector $x \subset X$ f is said to be non-dominated regarding a set

$$A \subseteq X \; f \; iff \; f \nexists \; a \in A: a > x \tag{2.8}$$

x is said to be Pareto optimal iff x is non dominated regarding X f .

In Figure 2.8, (Wikipedia, 2013) as for that given example the smaller values are better than larger values, points A and B are non dominated by any other, and C is dominated by both A and B.

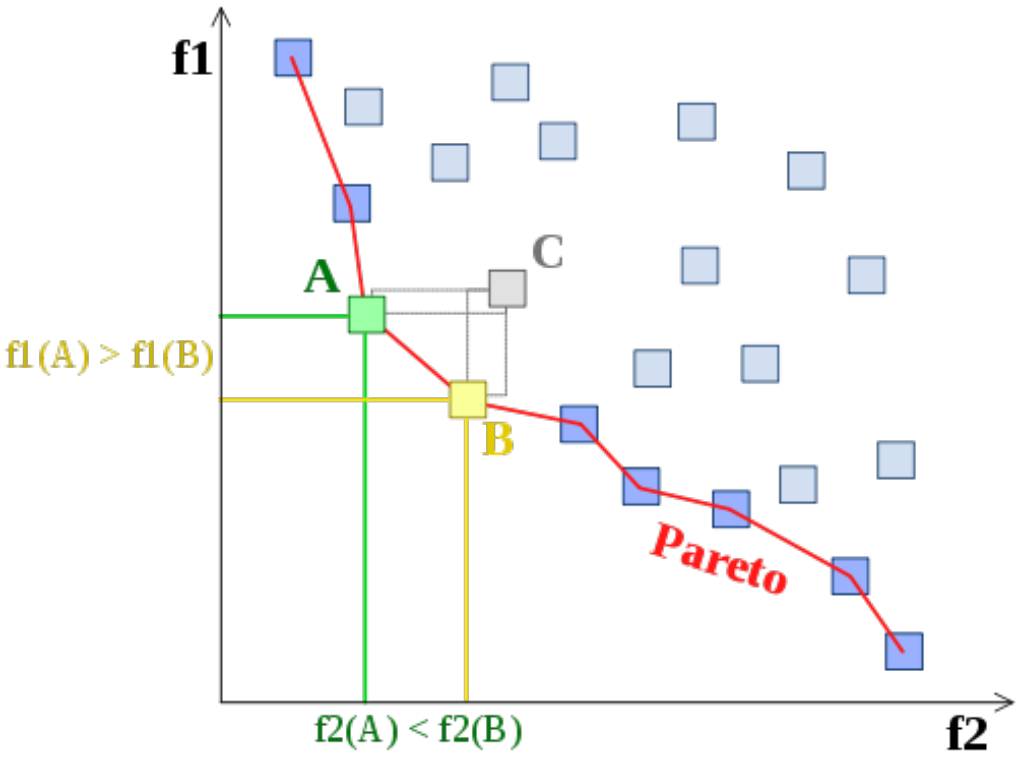


Figure 2.8 Example of Pareto frontiers.

SegGen algorithm uses Pareto optimality; it uses an external archive $\bar{P}$ to keep the non-dominated individuals with reference to both criteria and a current population P_t as

illustrated in Figure 2.9. Individuals selected from these two populations to produce new generations due to genetic operations. The new generation individuals substitute the current population and are used to update $\bar{P}$. At the end of the entire iterations, a set of potential results in the external archive $\bar{P}$.

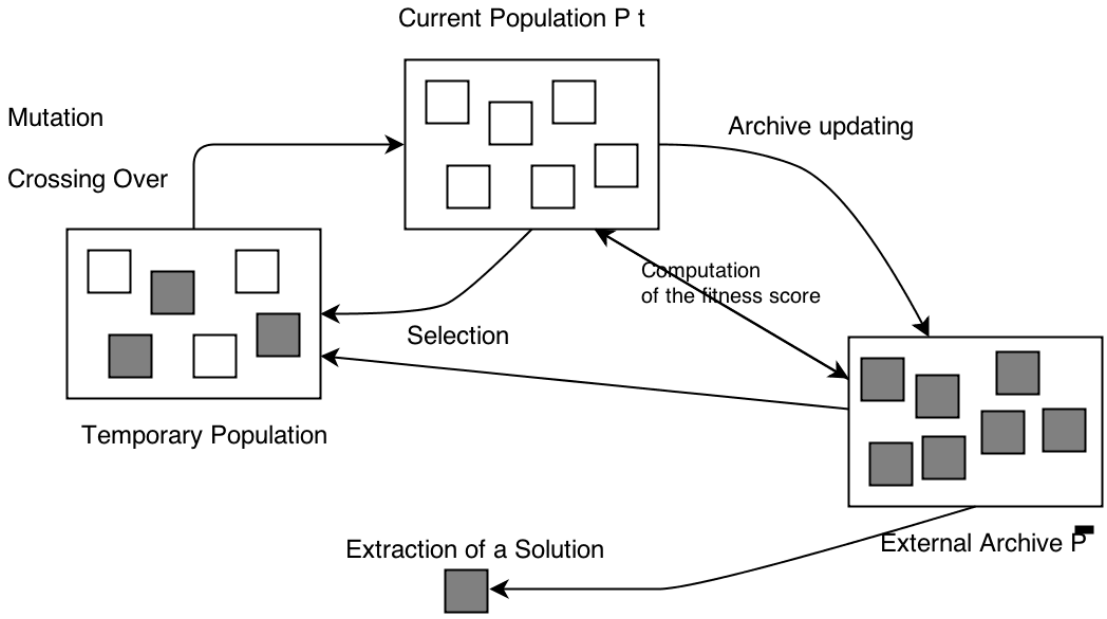


Figure 2.9 General flow of SegGen.

SegGen represents chromosomes as binary vector that means there are "1"s and "0"s, if $x_i = 1$ there is a boundary between sentence i and i+1, else there is no boundary between these sentences. The optimization objectives of SegGen are internal cohesion of segments $C(\vec{x}) \in [0,1]$ and dissimilarity between adjacent segments $D(\vec{x}) \in [0,1]$. SegGen formulates its optimizer as following,

$$O = \{\, \overrightarrow{x} \in \{0,1\}^{ns-1} \mid \nexists\, \overrightarrow{x'} \in \{0,1\}^{ns-1}, \\ ((\, C(\vec{x}) < C(\overrightarrow{x'})) \wedge \left(D(\vec{x}) < D(\overrightarrow{x'})\right)\} \tag{2.9}$$

Internal cohesion of segments: As mentioned previous section, SegGen uses vector representation of the sentences and uses similarities between sentences with regard to cosine measure.

$$Sim(s_1, s_2) = \frac{\sum_{i=1}^{t} w_{i,s_1} \times w_{i,s_2}}{\sqrt{\sum_{i=1}^{t} w_{i,s_1}{}^2} \times \sqrt{\sum_{i=1}^{t} w_{i,s_2}{}^2}} \quad (2.10)$$

SegGen introduces formula of internal cohesion of segments with using above cosine similarity formula with number of segments of the individual nseg, the segment i of the individual seg_i, sum of the sentence similarities of the segment seg_i, and the number of possible couples of sentences in the segment is Ncouples.

$$C(\vec{x}) = \frac{1}{nseg} \times \sum_{i=1}^{nseg} \frac{SumSim(seg_i)}{Ncouples(seg_i)} \quad (2.11)$$

Dissimilarity between adjacent segments: SegGen firstly computes dissimilarity between two segments as follows:

$$SimSeg(seg_1, seg_2) = \frac{\sum_{s_j \in S(seg_1)} \sum_{s_k \in S(seg_2)} Sim(s_j, s_k)}{|S(seg_1)| \times |S(seg_2)|} \quad (2.12)$$

a segment seg_i, a sentence s_j, the set of the sentences of the segment S(seg_i), and the cardinality of S(seg_i) is |S(seg_i)|.

After calculation dissimilarity between two segments, SegGen can define final dissimilarity between adjacent segments in an individual following:

$$D(\vec{x}) = 1 - \left(\frac{\sum_{i=1}^{nseg} SimSeg(seg_i, seg_{i+1})}{nseg-1} \right) \quad (2.13)$$

The algorithm uses this computation of fitness values in order to select applicant individuals for genetic operators. SegGen comprises ordinary genetic operators such as *roulette wheel* selection which the fittest individual has a greater chance of selection

than weaker one, classical single point crossover and a mutation with a probability shifts a “1” to the next or previous position in the individual representation.

Hardness value of the individual in the external archive: Algorithm uses a hardness value for each individual of the external archive in order to pick individuals from the external archive. Hardness value H denotes a number that an element x € $\bar{P}$ dominates the number of individuals of current population on both objective functions.

$$H(\vec{x}) = \frac{|\{\vec{y} \mid \vec{y} \in P_t \wedge C(\vec{x}) \geq C(\vec{y}) \wedge D(\vec{x}) \geq D(\vec{y})\}|}{|P_t|+1} \tag{2.14}$$

Then, the fitness value of x € $\bar{P}$ is inverse of its hardness value.

$$F(\vec{x}) = \frac{1}{H(\vec{x})} \tag{2.15}$$

Fitness value of an individual: On the other hand, the fitness value of an individual in the current population can be defined as sum of the hardness value of individual’s y dominating x:

$$F(\vec{y}) = \frac{1}{1+\sum_{\vec{x} \in \bar{P} \wedge C(\vec{x}) \geq C(\vec{y}) \wedge D(\vec{x}) \geq D(\vec{y})} H(\vec{x})} \tag{2.16}$$

The main flow of SegGen algorithm is given in Figure 2.10. First of all, the program represents individuals as binary vectors and initializes the population and an empty external archive. After the initialization of population, if the loop iterator at the first generation, the program picks out Pareto frontiers of the population and copies them into the external archive. If the loop iterator at the later generations, the program evaluates the fitness value of individuals in accordance with the fitness calculation examined previously. After the fitness evaluation step, the program applies selection, crossover and mutation operators to the current population and picks out new non-dominated individuals and updates the external archive. The program takes into account this new population that is produced by selection, crossover and mutation

operators in order to sustain the current generation. When algorithm meets the stop criterion which stagnation of the population evaluation, the external archive has several potential results. The algorithm selects the best result from the external archive.

```
SegGen
individual <- represents as binary vector
Pt <- init population
P' <- init external archive (not populated)
while stop criteria not yet encountered do
      if first generation then
            x' <- select pareto frontiers of population
            P' <- x' copy them into external archive
      else
            Eval F(x)U F(x') <- eval fitness all individuals
            x <- apply selection, crossover and mutation
            Update P' <- update external archive
end
x' <- select the best individual from external archive
return x'
```

Figure 2.10 Pseudocode of SegGen.

Due to the fact that all documents do not require the same number of generations to reach a satisfying segmentation; stop criterion of the algorithm is inertia of the population evaluation. When the algorithm meets the stop criterion, the external archive contains many potential segmentation and we have to extract best segmentation from this potential result set. Extraction of solution is explained in section 4.1.

3 PROPOSED IMPROVEMENT APPROACH TO SEGGEN ALGORITHM

As mentioned previous chapter, SegGen is a text segmentation method that benefits from genetic algorithm to solve bi-criteria optimization problem. However, the genetic algorithm used by SegGen is exact a generic type of genetic algorithm. In other words, it does not contain meaning peculiar to text segmentation.

3.1 Motivation for Improvement Approach to SegGen Algorithm

This thesis describes improvements that have been implemented in the approach taken by SegGen by tuning the genetic algorithm parameters according with the evolution of the quality of the generated populations. Two kinds of reasons originate the tuning of the parameters and have been implemented here. First as it could be measured by the values of global criteria of the population quality, the global quality of the generated populations increases as the process goes and it seems reasonable to set values to parameters and define new operators, which favor intensification and diminish diversification factors in the search process. Second since individuals in the populations are plausible segmentations it seems reasonable to weight sentences in the current segmentation depending on their distance to the boundaries of the segment they belong to for the calculus of similarities between sentences implied in the two criteria to be optimized. A brief justification of this last point, is that the more we can trust on a segmentation, the more we can take into account the importance of a sentence in the similarity of segment, depending on its position with regards to the boundaries of the segment it belongs to.

3.2 Genetic Operators Tuning

The main aim of genetic algorithms is finding a solution to complex problems by a method is inspired the process of evolution in nature. Nature evolves creatures which are best designed to correspond their environments by selecting features is called survival of the fittest. Following this approach, genetic algorithms perform by combining potential solutions to a problem together in a way is disposed to produce better solutions over sequential generations. Genetic algorithms are one form of local search that starts from initial configuration and makes evolution-based changes to the configuration until reaching the goal. Since the methods are attempting to optimize a set of objectives but will mostly find local maxima rather than a global maximum, local search methods are also known as local optimization (Coppin, 2004).

Researchers usually accomplish a proper balance between exploration and exploitation ability in searching or optimization algorithms. Exploration means searching search space as much as possible, while exploitation means concentrating on one point as usual global maxima. Specific to GA, crossover operators are widely used to lead population to converge on the good solutions and mutation operators are mostly used to provide exploration.

3.2.1 Mutation Operator Tuning

Mutation changes one or more gene values or position in a chromosome from its initial state. The goal of the mutation operator is to maintain genetic diversity from present generation to next generations to inject new solutions into the population. The departing point is the fact that mutation causes random changes on individuals. As new and differing individuals join the population, increasing diversity of the population offers a chance to reach more qualified individuals. A genetic algorithm is generally terminated when it converges. Convergence occurs when most of the individuals in a population have very similar genetic content. When convergence may happen very rapidly so that it becomes impossible to reach to the goal a problem occurs, which is

called premature convergence. Hence, using mutation allows avoiding the premature convergence problem. The effect of mutation can be visualized in the following way:

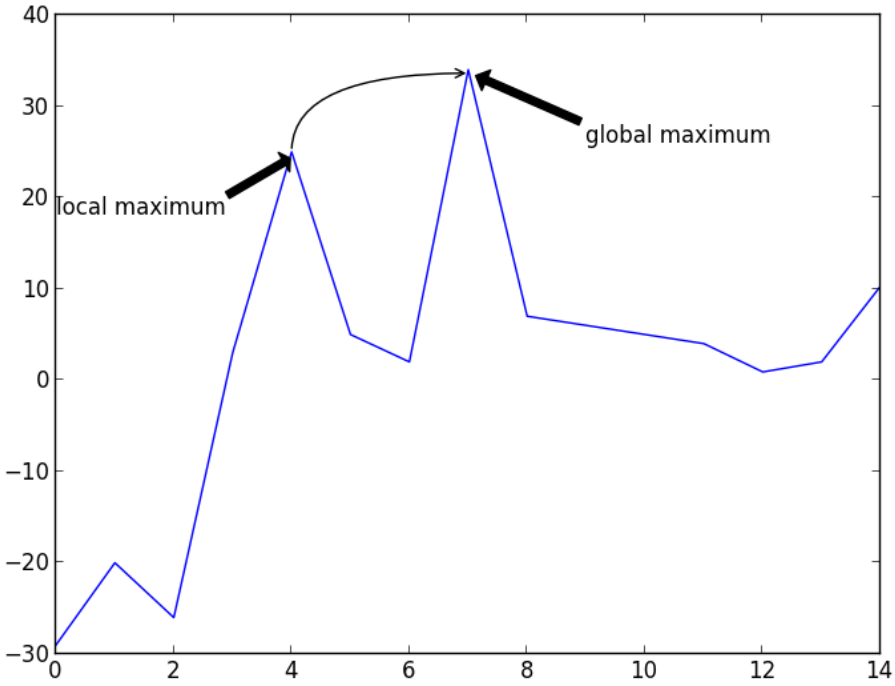


Figure 3.1 The Effect of Mutation Operator.

Assume that there is a function in Figure 3.1. The aim is to reach to the highest location is called global maximum of the function. The algorithm can be stuck on a higher location is called local maximum and terminated there. A successful mutation can create completely random solutions, leading to dislocate from the local maximum and can have a chance to find its way to the global maximum (Arslanoglu, 2006). Obviously, the algorithm prefers more exploitation at the end of search process to ensure the convergence of the population to the global optimum. Consequently, mutation provides doing its best to avoid premature convergence and explore more areas.

Due to the powerful effects of mutation operator, proposed mutation operator tuning approach attempts to create a specific interpretation for text segmentation. As

mentioned previous chapter, SegGen represents the individuals of population as binary vectors that means there are "1"s and "0"s, if x_i = 1 there is a boundary between sentence i and i+1, else there is no boundary between these sentences. A boundary indicates that a thematic changing occurs between adjacent sentences. Thus, the interpretation of an individual as a potential segmentation seems more appropriate than only bit string representation. As a consequence, the proposed tuning approach manipulates segmentation boundaries by mutation operator.

```
mutation_add_boundary
input: individual
       Pmut <- mutation probability
output: mutant individual
process:
      p <- random probability value
      if p <= Pmut then
            indices <- determine all "0" indices
            selected <- select a random index
            individual[selected] <- "1"
      end
return individual
```

Figure 3.2 Pseudocode of add boundary mutation

Proposed tuning approach includes two types of mutation in addition to the mutation types used by SegGen. First, it adds a boundary into the selected individual as defined pseudo code of mutation in Figure 3.2. Our new mutation operator can change the existing of boundary in the selected individual. We can effectively use the boundary adding by new mutation for getting the accurate result in mutation search.

The second type of proposed mutation is based on with a probability Pmut that shifts selected boundary to next sentence on given individual. It shifts the two selected

boundaries to next sentences. It simply shifts the two selected “1” bites to the next position in the individual vector as seen in Figure 3.3. Moreover, if selected one is a qualified individual, cost of the shifting boundary process is smaller than recreating qualified individual.

```
mutation_shift_boundary
input: individual
       Pmut <- mutation probability
output: mutant individual
process:
      p <- random probability value
      if p <= Pmut then
            indices <- determine all "1" indices
            if size(indices)>= 2 then
                  selected <- select two random indices
                  individual[selected1] <- "0"
                  individual[selected1+1] <- "1"
                  individual[selected2] <- "0"
                  individual[selected2+1] <- "1"
            else if size(indices) == 1 then
                  selected <- select two random indices
                  individual[selected] <- "0"
                  individual[selected+1] <- "1"
            end
      end
return individual
```

Figure 3.3 Pseudocode of mutation shift boundary.

3.2.2 Mutation Probability Tuning

In GA, mutation operators are mostly used to provide exploration. According to mutation facts, too small mutation rate can cause to premature convergence that means getting stuck on local maximum instead of global maximum. On the other hand, too high mutation rate enhances the probability of searching more areas in searching problem space. At the same time, it interferes with population to reach to any optimum solution and throws the solution into far distances of current solution. Due to the searching algorithm acts different exploration and exploitation ability in different stage of the search process, the best value of mutation is special to problem. Thus, Gaspar (2010) indicates that a more dynamic mutation rate is more preferred. More complex algorithms to adaptively tune the mutation rate according to the problem and the situation of the current population compared to the previous generation.

Following the approach, we change probabilities of mutation in general and more specifically. In this study, we will change the mutation probability so that we can release the algorithm if it sticks to the local maxima so it can jump to the global maximum. The departing point is the fact that mutation causes random changes on individuals by its nature and larger mutation rates to tend high genetic variety and avoid local maxima. As new and differing individuals join the population, increasing diversity of the population offers a chance for reaching more qualified individuals. There is low probability of mutation in early generations of the program. In subsequent generations, mutation probability is either increased or stabilized taking into consideration average quality of the population. If the program is close to the goal, we may be confident on the boundaries; if not we have to diversify the solution. In this way, we increase the possibility of reaching the goal by increasing the mutation probability.

3.2.3 Crossover Operator Tuning

Crossover is a genetic algorithm operator that recombines two chromosomes to produce two new chromosomes. Crossover operators are common to lead population to

converge on a specific point in landscape. In GA, more exploitation at the end of search process are preferred because, search process wants to ensure the convergence of the population to the best solution. In contrast with mutation, crossover does not recombine not existed gene pair in the population. From this point of view, we have implemented two types of custom crossover. On the other hand, we have added a different type of crossover that similar to mutation operator.

First type of crossover operator is a multipoint crossover instead of uniform mono-point crossover operator as illustrated in Figure 3.4. We use two common boundary points of selected parents because the generated individuals have to keep existing boundaries on some part of the document to be defined. Due to the keeping existing boundaries in crossover operation, tuning crossover operator process provides a multipoint crossover more specific than ordinary multipoint crossover.

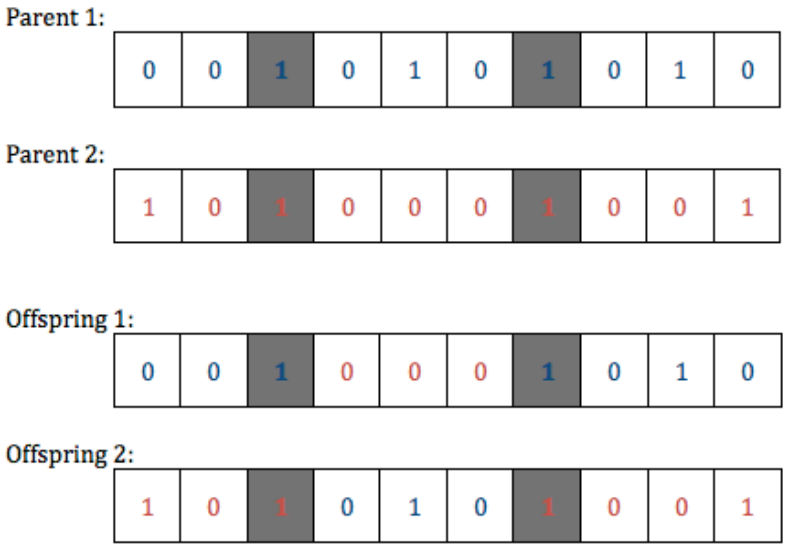


Figure 3.4 Proposed multipoint crossover.

Second type of crossover operator can be described as keeping the same number of boundaries. The facts that parent individuals that selected for crossover operator have similar number of boundary give a clue about the similarity of the two chosen

individuals. The pseudo code of the second type proposed crossover operator as shown in Figure 3.5. It determines the positions of the part of boundaries and compares the positions, if the position difference is equal or smaller than threshold value; the operator cut and interchanges the parents at the defined position. Thus, relying on similar number of boundaries indicates a chance of reach the goal.

```
crossover_same_boundary_count
input: parent1
       parent2
output: offspring1
        offspring2
process:
      b_count1 <- 40% of the number of boundaries of first parent
      b_count2 <- 40% of the number of boundaries of first parent
      position1 <- position of b_count1
      position2 <- position of b_count2
      if |position1-position2| <= 3 then
            offspring1 <- cut and interchange parents
            offspring2 <- cut and interchange parents
      end
return offspring1, offspring2
```

Figure 3.5 Pseudocode of proposed second type crossover operator.

Third type of proposed crossover is merge parents and eliminates exceeding boundaries. Parents are entered into a logical OR operation, after this operation keeping average number of boundary and eliminating exceeding boundaries. So these all approaches have the same objective that there is not only a binary bit sequence, but also there is a potential segmentation and the algorithm considers specific modifications of genetic algorithm in segmentation context by obtaining boundaries.

3.3 Fitness Function Tuning

Fitness function is the objective function of the genetic algorithms that it used to measure qualification of given individual as a solution to problem by analyzing its genetic content value. It has an important role in main mechanism of genetic algorithm, because the algorithm decides to select individuals to the next generations by its fitness value. As explained previous chapter, the objectives of SegGen are maximum values of internal cohesion of segments and dissimilarity between adjacent segments. SegGen benefits from results of its objective functions to determine non-dominated individuals in the population.

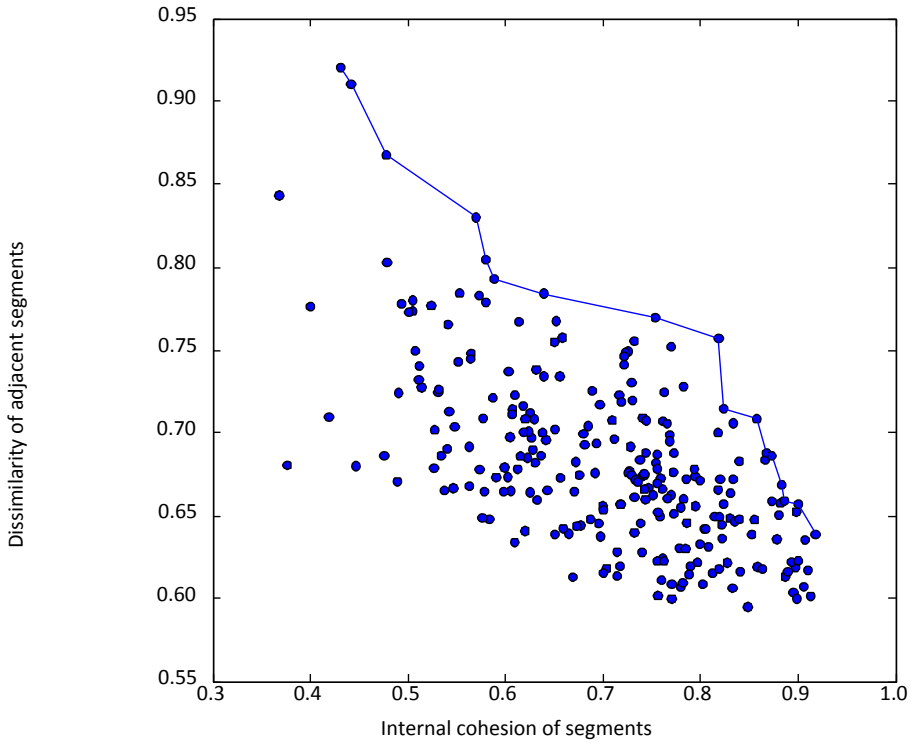


Figure 3.6 Example of Pareto frontiers produced by SegGen.

Proportion of the number of non-dominated individuals in the population indicates fitness values of the individuals. As shown in Figure 3.6, our algorithm wants to find maximum values of the similarity of internal cohesion and dissimilarity between

adjacent segments, so we can say points that lie on the Pareto frontier line are non dominated by any other, and smaller value points are dominated by frontier points.

On a given segmentation, the similarity measures between sentences have to be changed to give different weights to sentences depending on their proximities with a boundary but this has to take place during the ongoing process. At the beginning of the process we could only have a low confidence on the position of the boundaries and so have no reason to be treated differently from other sentences in the calculus of similarity, with regard to our study, the calculus of cohesion of segments and dissimilarity between adjacent segments. On subsequent processes, quality of populations increases and we may reasonably think that boundaries are roughly in their final position. So we can be more confident and take this into account in the calculus of similarity. The segmentation in the current population is of a better quality as population evolves but, there is no reason to think that boundaries a more or less in their final position. So since this is near the boundaries that thematic changes occurs, cohesion has to be measured with less or no influences of them. Therefore, the idea of tuning the fitness function came into view. Our proposed method takes into account the nature of the coding of individuals, which in this case are segmentation instances, represented by binary vectors corresponding to the positions of the boundaries of the segmentations. As shown in Figure 3.7, there is the creation of a weighted value vector of given individual. We gave different importance value to sentences depending on their positions with regards to the boundaries of the segment they belong to. We figured out that near the boundaries and adjacent sentences of the near boundaries have logarithmic importance values (negative values). Thus, we created weighted factor individuals. Due to thematic changes occurs near the boundaries, weighted factor evaluation process provides that these boundary points have less importance on the calculus of cohesion of segments and dissimilarity between adjacent segments.

```
weighted_value_vector
input: individual
output: weighted value vector
process:
      w <- the vector only contains 1
      for i=0 to size(individual)-1 do
            if individual[i] == 1 then
                  w[i-1],w[i+2] <- -log4
                  w[i],w[i+1] <- -log3
            end
      end
return w
```

Figure 3.7 Pseudocode of creation of the weighted value vector

4 EXPERIMENTAL RESULTS OF PROPOSED METHOD

This section consists of the experimental process of the proposed improvement approach to SegGen. This process comprises of solution retrieval, preparation of input text, measure method, results of the implemented program, and stability test.

4.1 Solution Retrieval

As mentioned Section 2.2.2, at the end of the SegGen algorithm operation, because of the external archive contains more than one potential segmentations and we have to extract best segmentation from this potential result set, result retrieval requires a few additional processes. The best solution retrieval form potential segmentation results can be performed by an aggregation function. For example, a linear aggregation of both objective functions as following,

$$Agg(\vec{x}) = C(\vec{x}) + \alpha \times D(\vec{x}) \tag{4.1}$$

The coefficient α weights the second objective compared to the first.

We used aggregation method of SegGen, thus we extracted best aggregation score of individual from potential result set. In the extraction process, we consider aggregation evaluation in experimental studies of SegGen and α is obtained around 5. Then, we select aggregation score greater than 4.9. After this filter process we choose best score of filtered result set. On the other hand, aggregation score of weighted fitness function does not provide sufficient selectivity in the extraction process. We distinguished a negative correlation between aggregation score of basic fitness function and aggregation score of weighted fitness function. Since this negative correlation we decided on a threshold value in order to use select the best result.

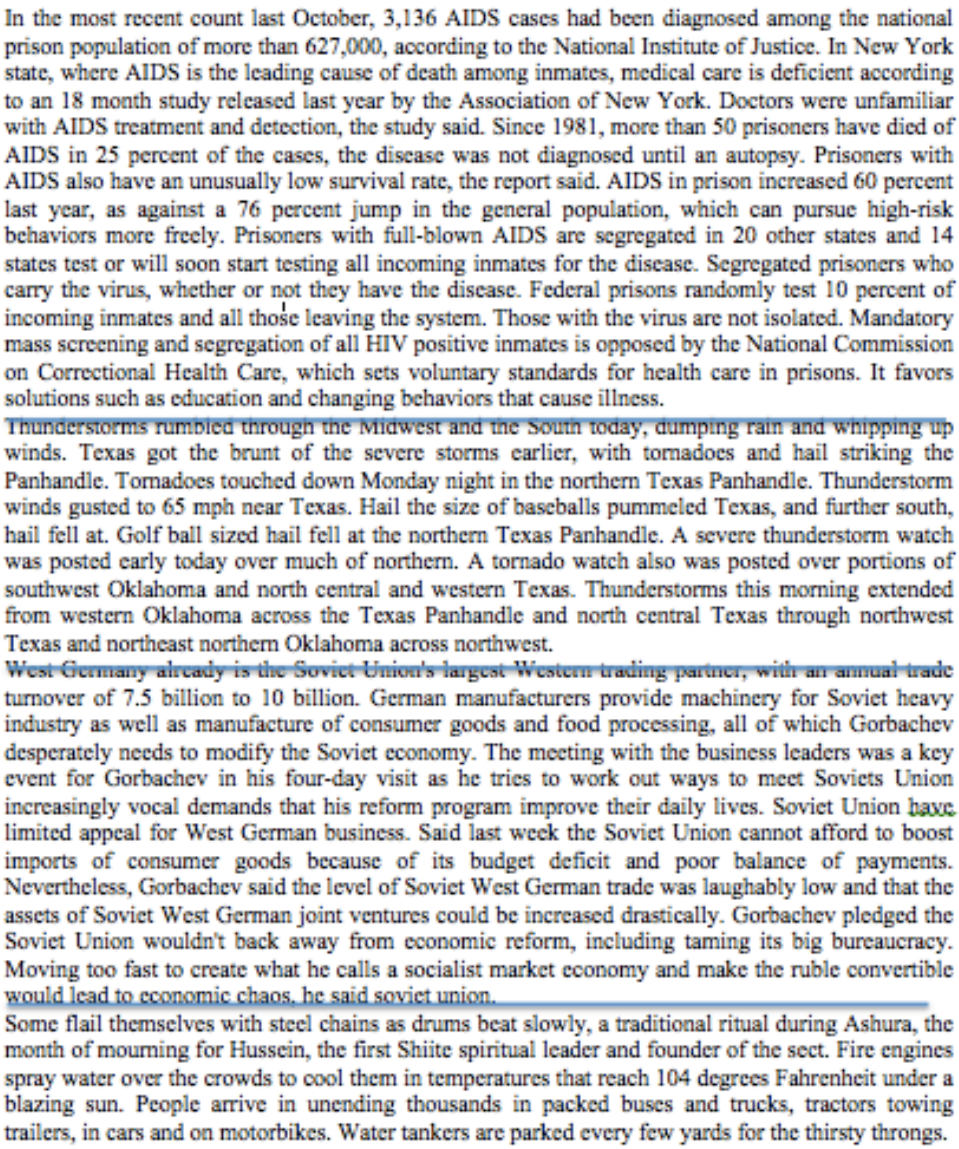
4.2 Experimental Process

We used test texts, which consist of articles from the Associated Press published all the year around 1989 (Harman, 1993). As shown in Figure 4.1, we concatenated sample articles which have various topics, selected from set of 350 documents. Due to the subjectivity of the task this type of corpus is generally used by the community, we decided to follow the same method. We created several corpora in order to be used in the experimental process noted as T(ns,nb) where ns is number of sentences and nb is the average number of boundaries. The test corpora are A(30,2), B(30,2), C(38,5), D(50,5), and E(55,7).

We used a criterion *WindowDiff* (Pevzner & Hearst, 2002) that is a metric commonly used in text segmentation, as an evaluation metric. WindowDiff indicates difference between reference segmentation and the method to be evaluated. It considers the number of boundaries between two sentences separated from a distance k, as shown in formula,

$$Windiff(hyp, ref) = \frac{1}{N-k} \sum_{i=0}^{N-k} (|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})|) \qquad (4.2)$$

$b(x_i, x_j)$ is the number of boundaries between i and j in a segmented text x which consists of N sentences, ref points to the segmentation of reference and hyp the one found with the method to evaluate. We chose the average number of size of segments as the value of k.



In the most recent count last October, 3,136 AIDS cases had been diagnosed among the national prison population of more than 627,000, according to the National Institute of Justice. In New York state, where AIDS is the leading cause of death among inmates, medical care is deficient according to an 18 month study released last year by the Association of New York. Doctors were unfamiliar with AIDS treatment and detection, the study said. Since 1981, more than 50 prisoners have died of AIDS in 25 percent of the cases, the disease was not diagnosed until an autopsy. Prisoners with AIDS also have an unusually low survival rate, the report said. AIDS in prison increased 60 percent last year, as against a 76 percent jump in the general population, which can pursue high-risk behaviors more freely. Prisoners with full-blown AIDS are segregated in 20 other states and 14 states test or will soon start testing all incoming inmates for the disease. Segregated prisoners who carry the virus, whether or not they have the disease. Federal prisons randomly test 10 percent of incoming inmates and all those leaving the system. Those with the virus are not isolated. Mandatory mass screening and segregation of all HIV positive inmates is opposed by the National Commission on Correctional Health Care, which sets voluntary standards for health care in prisons. It favors solutions such as education and changing behaviors that cause illness.

Thunderstorms rumbled through the Midwest and the South today, dumping rain and whipping up winds. Texas got the brunt of the severe storms earlier, with tornadoes and hail striking the Panhandle. Tornadoes touched down Monday night in the northern Texas Panhandle. Thunderstorm winds gusted to 65 mph near Texas. Hail the size of baseballs pummeled Texas, and further south, hail fell at. Golf ball sized hail fell at the northern Texas Panhandle. A severe thunderstorm watch was posted early today over much of northern. A tornado watch also was posted over portions of southwest Oklahoma and north central and western Texas. Thunderstorms this morning extended from western Oklahoma across the Texas Panhandle and north central Texas through northwest Texas and northeast northern Oklahoma across northwest.

West Germany already is the Soviet Union's largest Western trading partner, with an annual trade turnover of 7.5 billion to 10 billion. German manufacturers provide machinery for Soviet heavy industry as well as manufacture of consumer goods and food processing, all of which Gorbachev desperately needs to modify the Soviet economy. The meeting with the business leaders was a key event for Gorbachev in his four-day visit as he tries to work out ways to meet Soviets Union increasingly vocal demands that his reform program improve their daily lives. Soviet Union have limited appeal for West German business. Said last week the Soviet Union cannot afford to boost imports of consumer goods because of its budget deficit and poor balance of payments. Nevertheless, Gorbachev said the level of Soviet West German trade was laughably low and that the assets of Soviet West German joint ventures could be increased drastically. Gorbachev pledged the Soviet Union wouldn't back away from economic reform, including taming its big bureaucracy. Moving too fast to create what he calls a socialist market economy and make the ruble convertible would lead to economic chaos, he said soviet union.

Some flail themselves with steel chains as drums beat slowly, a traditional ritual during Ashura, the month of mourning for Hussein, the first Shiite spiritual leader and founder of the sect. Fire engines spray water over the crowds to cool them in temperatures that reach 104 degrees Fahrenheit under a blazing sun. People arrive in unending thousands in packed buses and trucks, tractors towing trailers, in cars and on motorbikes. Water tankers are parked every few yards for the thirsty throngs.

Figure 4.1 An example of produced input text.

4.3 Result Comparison

Result comparison section consists of two main types of experimental processes. The main difference between two experimental processes is heterogeneity of the population. First type of population consists of individuals that have various numbers of boundaries. The second type of population has individuals that only given number of boundaries.

4.3.1 Various Numbers of Boundaries

On implementation of this type of population, the algorithm creates random number of boundaries while initialization of the population. Three different groups are used in experimental process. First group consists of five versions of algorithm. These are:

- *Basic:* The basic version of the algorithm.
- *M1:* We previously mentioned about adding two types mutations into algorithm. This version of the algorithm comprises of added two new mutations into basic version.
- *M2:* This version consists of changing mutation probability besides two new mutation types.
- *C:* The C version has tuned crossover operator.
- *M2C:* The version comprises applying combination of new mutation types, changeable mutation probability and crossover.

The second group includes two versions of weighted fitness function changes besides basic type. These are:

- *Basic:* The basic version of the algorithm.
- *Weighted:* Weighted fitness function is applied in this version.
- *M2C-Weighted:* Weighted fitness function is applied besides combination of new mutation types, changeable mutation probability and crossover.

The last group is not quite different from other groups that it comprises mixed type of previous versions. With reference to first experimental results of combination of proposed methods, we predicted that a mixed and dynamic version of algorithm would produce satisfying results. Mixed-type algorithm works as follows: every ten generation, n individuals are randomly selected from population, and calculated population quality according to the aggregation values of selected individuals that can be determined an average quality of current generation. There is a trade-off between population quality and selected proposed method.

The third group includes three version of algorithm that are basic, mixed-type and weighted-mixed type.

- Basic: The basic version of the algorithm.
- Mixed: According to population quality, randomly selected proposed methods.
- Weighted- Mixed: Weighted fitness function changes besides mixed type.

The initial population size is 250 and following generations consists of about 100 individuals. The algorithm usually executes around 250 generations. Due to depending on random parameters by genetic operators of SegGen, each version of algorithm executes 10 times on each corpus, and then best results are extracted from results.

First results of this experimental study of the algorithm obtained on the evaluation corpora are promising. Recall that the lower values are better, because Windiff indicates difference between reference segmentation and the method to be evaluated. Even if the suiting of the parameters of the algorithm currently builds upon empirical values, in the Table 4.1, tuned genetic operator versions of the algorithm results seem better than results of basic version of the algorithm. Especially, combination of all proposed tuning approaches is often better than single versions.

Table 4.1 Basic SegGen and Tuned Genetic Operators

	Basic	M1	M2	C	M2C
A(30,2)	31.7	30.9	21.7	31.8	27.4
B(30,2)	28.2	37.0	29.0	20.3	20.1
C(38,5)	36.4	37.4	36.5	32.9	33.4
D(50,5)	52.5	53.0	52.4	51.3	47.5
E(55,7)	61.3	52.8	52.3	50.0	37.8

Regarding the results in Table 4.2, calculation of weighted value of sentences in accordance with their position in the whole text, also is promising. Using weighted value method with tuned genetic operators will be better, because these results are first empirical results and tuning of genetic operator process uses random parameters. But it

shows that the approach still needs some improvement such as tuning the weighted factor individual.

Table 4.2 Basic SegGen and Tuned Fitness Function

	Basic	Weighted	M2C-Weighted
A(30,2)	31.7	26.0	33.0
B(30,2)	28.2	22.0	32.1
C(38,5)	36.4	38.2	37.0
D(50,5)	52.5	60.6	55.0
E(55,7)	61.3	57.3	61.4

According to the results in the Table 4.3, mixed and dynamic adjustment of the proposed approaches is better than previous suggestions. The mixed algorithm takes into account global quality of population.

Table 4.3 Basic SegGen and Mixed Types

	Basic	Mixed	Weighted-Mixed
A(30,2)	31.7	24.6	28.1
B(30,2)	28.2	22.3	29.2
C(38,5)	36.4	25.5	38.5
D(50,5)	52.5	50.1	54.4
E(55,7)	61.3	48.0	54.1

Weighted type of the algorithm needs some improvement, because the improved implementation of the algorithm can provide a qualified operation that assigning value to sentences regarding with their positions in the text. This proposal is expected to taken better results than basic algorithm. The first experimental results of proposed weighted fitness function approach are not satisfying. We figured out that there are two reasons of these results. First, input texts are comprises of artificial texts, which do not contain linguistic indices. Second, the first experimental segment lengths are short in text inputs. For example, the average segment length of the E(55,7) corpora is 8

sentences. Weighted fitness function proposal gives negative importance values to near the boundaries and adjacent sentences of the near boundaries. The shorter length of segments does not provide a fair distribution of importance value. By this way, we prepared another corpora F(98,4) with an average segment length is 20 sentences.

Table 4.4 Weighted results of long segment size input

	Basic	Weighted	M2C-Weighted	Mixed-Weighted
F(98,4)	78.2	75.3	72.7	70.8

As shown in Table 4.4, weighted fitness function approach within long segment size input text gives better results than weighted fitness function approach within short segment size inputs. Additionally, M2C and Mixed type of the algorithm gives better results than basic type of the algorithm.

To sum up, the parameters of the algorithm in first results rest upon empirical values, first results are promising. We are convinced they will be better by automatically fixing the values of the various parameters in using an automatic tuning method instead of the empirical guess we have done in the current state of this research.

1.1.2 Fixed Numbers of Boundaries

Separation of given text into semantically coherent pieces is called thematic segmentation. During the thematic boundary identification process, the thematic boundaries between different subjects are determined. Thematic segmentation is subjective. Even though thematic segmentation process which is performed by a human, does not always give the perfect result. A person can determine ten boundaries on given text; another person can be determined four boundaries on the same text. By reason of subjectivity of text segmentation in our experimental studies, the fixed numbers of boundaries algorithm wants to keep individuals, which has only fixed

numbers of boundaries. Thus a new parameter is added to the algorithm: the number of boundary. When population initializes, the population contains individuals with a given number of boundaries.

The same groups are used in the previous experimental process. First group consists of five versions of algorithm. The second group includes two versions of weighted fitness function changes besides basic type. The third group includes three version of the algorithm that are basic, mixed-type and weighted-mixed type.

Table 4.5 Results of Fixed Numbers of Boundaries

	Basic	M1	M2	C	M2C
A(30,2)	19.2	19.1	18.5	18.9	17.9
B(30,2)	20.1	21.3	20	20.5	18.7
C(38,5)	24.3	20.8	22.8	23.5	22
D(50,5)	25.6	24.5	22.8	24.7	24
E(55,7)	46.6	42.5	40.8	45.7	41.5

Regarding to Table 4.5, first results of fixed numbers of boundaries are better than results of various numbers of boundaries. Both the basic type of the algorithm and the types of applied genetic operator tuning results are lower than various numbers of boundaries, because beginning of the search process population does not irrelevant solutions, so this trick makes the process even better.

Table 4.6 Results of Fixed Boundaries of Basic and Weighted types

	Basic	Weighted	M2C-Weighted
A(30,2)	19.2	20.1	19.7
B(30,2)	20.1	22	21.5
C(38,5)	24.3	26.2	27
D(50,5)	25.6	25	26.3
E(55,7)	46.6	45.4	47.1

As shown in Table 4.6, the results of basic type of the algorithm and the types of algorithm, which tuned fitness function, are placed. The weighted types of the algorithm sometime gives better results than the various numbers of boundaries results, but they need still improvement.

To sum up with results in Table 4.7, applied mixed genetic operators of the algorithm is better than other types of the algorithm.

Table 4.7 Results of Fixed Numbers of Boundaries Basic and Mixed Types

	Basic	Mixed	Weighted-Mixed
A(30,2)	19.2	16.5	22
B(30,2)	20.1	16.8	21.5
C(38,5)	24.3	22.2	24.2
D(50,5)	25.6	23.8	26
E(55,7)	46.6	41.4	42.5

4.4 Stability Test

Evaluation metrics of text segmentation methods do not always give perfect results. The evaluation results are depending on the sensibility of the methods. As mentioned previous section, we used WindowDiff as an evaluation metric. We mentioned previously, our segmentation criteria are a high internal cohesion and a low similarity between adjacent parts. These two criteria do not consider the arranging of the sentences in the segments. If a method is good, it should gives the same results whatever the order of the sentences is in each segment (Lamprier et al., 2007).

So to evaluate the performance of our algorithm, we performed random sentence permutations inside each segment of previously mentioned corpora A(30,2), B(30,2), C(38,5), D(50,5), and E(55,7). Because of using concatenated passages that do not include linguistic indices, changes in the order of the sentences inside the each text parts should not affect in a negative way the determination of boundaries. After the

changes in the order of sentences inside each segment new input corpora are segmented with our algorithm.

Table 4.8 Stability Results of Basic SegGen and Tuned Genetic Operators

	Basic	M1	M2	C	M2C
A(30,2)	30.2	31.4	30.1	32.1	28.4
B(30,2)	29.1	28.9	31.6	31.0	27.2
C(38,5)	41.4	29.4	38.2	29.0	33.8
D(50,5)	52.2	48.0	57.1	57.8	53.7
E(55,7)	54.3	49.0	51.3	55.6	46.6

Regarding the Table 4.8, changing the order of the sentences inside each segment should give similar results with basic order of sentences in each segment. We use WindowDiff with the result given by the method as reference segmentation. In reference to Table 4.9, the stability test results of configuration of tuned fitness function give around the results of the basic order of sentences in each segment. Additionally, it shows the tuned fitness function approach needs more improvement.

Table 4.9 Stability Results of Basic SegGen and Tuned Fitness Function

	Basic	Weighted	M2C-Weighted
A(30,2)	30.2	30.0	31.3
B(30,2)	29.1	31.2	28.5
C(38,5)	41.4	42.6	41.6
D(50,5)	52.2	53.2	55.6
E(55,7)	51.3	54.2	51.8

On the other hand, regarding to Table 4.10, results of mixed type of tuned genetic operators seems promising and also all types are similar to results of basic improvement approach.

Table 4.10 Stability Results of Basic SegGen and Mixed Types

	Basic	Mixed	Weighted-Mixed
A(30,2)	30.2	26.8	28.2
B(30,2)	29.1	24.2	28.7
C(38,5)	41.4	27.2	37.0
D(50,5)	52.2	43.0	49.3
E(55,7)	51.3	39.6	54.0

As a result, the first results based on empirical values are promising and with respect to first results our proposed stability test results are also satisfied. Because we have an expectation about stability of the proposed approach that it should give similar results to results of the algorithm within basic configuration of the input corpora.

As shown in Figure 4.2, the input text has some ordinary news reports about on cases AIDS in prisons in USA and about thunderstorms in Texas. On the other side, in Figure 4.3, there is a proposed segmentation result on the same news report.

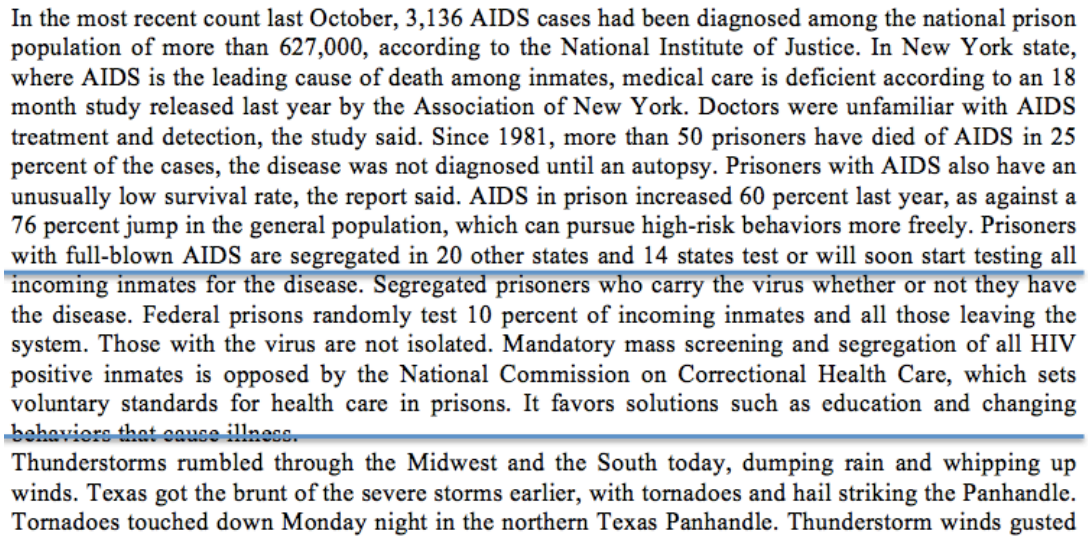
In the most recent count last October, 3,136 AIDS cases had been diagnosed among the national prison population of more than 627,000, according to the National Institute of Justice. In New York state, where AIDS is the leading cause of death among inmates, medical care is deficient according to an 18 month study released last year by the Association of New York. Doctors were unfamiliar with AIDS treatment and detection, the study said. Since 1981, more than 50 prisoners have died of AIDS in 25 percent of the cases, the disease was not diagnosed until an autopsy. Prisoners with AIDS also have an unusually low survival rate, the report said. AIDS in prison increased 60 percent last year, as against a 76 percent jump in the general population, which can pursue high-risk behaviors more freely. Prisoners with full-blown AIDS are segregated in 20 other states and 14 states test or will soon start testing all incoming inmates for the disease. Segregated prisoners who carry the virus, whether or not they have the disease. Federal prisons randomly test 10 percent of incoming inmates and all those leaving the system. Those with the virus are not isolated. Mandatory mass screening and segregation of all HIV positive inmates is opposed by the National Commission on Correctional Health Care, which sets voluntary standards for health care in prisons. It favors solutions such as education and changing behaviors that cause illness.

Thunderstorms rumbled through the Midwest and the South today, dumping rain and whipping up winds. Texas got the brunt of the severe storms earlier, with tornadoes and hail striking the Panhandle. Tornadoes touched down Monday night in the northern Texas Panhandle. Thunderstorm

Figure 4.2 An example of reference segmentation

The results sometimes do not be at the desired points, because we use composed of newspaper reports as input texts. Since usually more than one subtopic are combined in

a news text, for example, a report on cases AIDS in prisons in USA could consists of both AIDS and prisons subtopics. Due to the similarity of the two neighboring sentences is measured by the algorithm, the algorithm detected two subtopics that one topic is on the occurrence of the word related to AIDS and other topic is on the occurrence of the word related to prisons.


In the most recent count last October, 3,136 AIDS cases had been diagnosed among the national prison population of more than 627,000, according to the National Institute of Justice. In New York state, where AIDS is the leading cause of death among inmates, medical care is deficient according to an 18 month study released last year by the Association of New York. Doctors were unfamiliar with AIDS treatment and detection, the study said. Since 1981, more than 50 prisoners have died of AIDS in 25 percent of the cases, the disease was not diagnosed until an autopsy. Prisoners with AIDS also have an unusually low survival rate, the report said. AIDS in prison increased 60 percent last year, as against a 76 percent jump in the general population, which can pursue high-risk behaviors more freely. Prisoners with full-blown AIDS are segregated in 20 other states and 14 states test or will soon start testing all incoming inmates for the disease. Segregated prisoners who carry the virus whether or not they have the disease. Federal prisons randomly test 10 percent of incoming inmates and all those leaving the system. Those with the virus are not isolated. Mandatory mass screening and segregation of all HIV positive inmates is opposed by the National Commission on Correctional Health Care, which sets voluntary standards for health care in prisons. It favors solutions such as education and changing behaviors that cause illness.

Thunderstorms rumbled through the Midwest and the South today, dumping rain and whipping up winds. Texas got the brunt of the severe storms earlier, with tornadoes and hail striking the Panhandle. Tornadoes touched down Monday night in the northern Texas Panhandle. Thunderstorm winds gusted

Figure 4.3 An example of proposed segmentation result on subtopics

Consequently, the first results of the proposed approach to SegGen are promising. In detail, genetic operator tuning types give better results than the basic algorithm, but the genetic operators depends on randomness, so the algorithm performance is improved by the changing of types of used proposed genetic operator tuning.

Weighted fitness function proposes a reasonable improvement approach to the algorithm. The improvement proposed that after a period of generation creates more reliable individuals and gives different importance value to sentences depends on their positions that near the boundaries and adjacent sentences of the near boundaries have negative importance values. Weighted type of the algorithm needs some improvement, because the improved implementation of the algorithm can provide a qualified operation that assigning value to sentences regarding with their positions in the text. This proposal is expected to taken better results than basic algorithm. The first

experimental results of proposed weighted fitness function approach are not satisfying. We figured out that there are two reasons of these unsatisfying results. First, input texts are comprises of artificial texts, which do not contain linguistic indices. Because, we concatenated sample articles, which have various topics, selected from set of documents. Due to the lack of ordinary text layout, it does not benefit from linguistic clues in this case. Since this case, the study did not consider the linguistic indices in order to determine the thematic changing occurrences. Using real text in test would answer the purpose within tuned fitness function of the algorithm. Thus, there will be more confident fitness values of the individuals thanks to relying on linguistic indices. According to these confident individuals, creating weighted value vector of individuals would give more successful results. Second, the first experimental segment lengths are short in text inputs. For example, the average segment length of the E(55,7) corpora is 8 sentences. Weighted fitness function proposal gives negative importance values to near the boundaries and adjacent sentences of the near boundaries. The shorter length of segments does not provide a fair distribution of importance value.

We have an expectation about stability of the proposed approach that it should give similar results to results of the algorithm within basic configuration of the input corpora. Due to the lack of linguistic indices, two criteria of optimization do not into account the arranging of the sentences in the segments. Thus, changes in the order of the sentences inside the each text parts should not affect in a negative way the determination of boundaries. The results of stability test are similar to results of the algorithm within basic configuration of the input corpora, and validate the hypothesis about changing order of the sentences.

4.5 Extensions

This thesis would provide some extensions in the future. First of all, having the segmentation unit evolving during the process (evolving grain of segmentation during the process). Till now we consider sentences as unit to segment documents. It would be interesting to investigate other units, or having these units evolving during the

segmentation process. For example we can consider units as being k-contiguous sentences at the beginning of the processes and our problem becomes

I. Segmenting the text with units of a raw grain
II. Having grain evolving during process
III. Introducing linguistic indices at the end of the process to determine the exact position of the boundaries.

We suppose here that we stop the preceding (step 2) with units composed of several sentences.

Secondly, with another perspective, this method can be classified as a clustering method. The main aim of clustering is to group a set of objects in such a way that objects in the same cluster are more similar of each other than to those in other clusters. On the other hand, SegGen performs to find out the subtopics, which create internal coherence and are distinguished from other parts of the text. By this way, the significant similarity can be seen that between the main aim of clustering and the general approach. Due to the main aim of the algorithm is separating the text according to the their topics, this process is basically a clustering process.

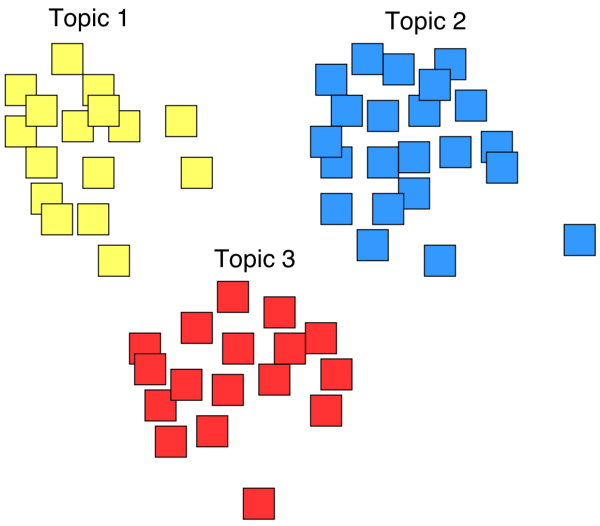


Figure 4.4 SegGen illustration as clustering method

As shown in Figure 4.4, SegGen separates given input texts with respect to topic of each other. Due to the maximum value of the internal consistency of each clusters and

the maximum value of dissimilarity of different clusters, we can say that SegGen acts like a clustering method. Although, there is a different point between clustering methods and SegGen that SegGen takes into account the order of the sentences. SegGen intends to separate the text into meaningful homogeneous at the boundaries of different subjects.

Finally, in this thesis, the first results based on empirical values, so a new different study will be better by automatically fixing the values of the various parameters in using a kind of learning method.

5 CONCLUSION

Automatic text segmentation identifies the most important thematic breaks in texts by setting the boundaries between segments on the ground of some given criteria, such as the internal cohesion of so determined segments and the dissimilarity between adjacent segments. Contrary to most of existing algorithms that create boundaries sequentially and set the boundaries between segments on local criteria, SegGen algorithm permits to take a decision on the base whole text to be segmented since all the boundaries between potential segments are set at the same time.

In this thesis, we presented our improvement approaches to SegGen algorithm, which consists of tuning genetic operators and tuning fitness function. We have presented the first results of an ongoing work aiming at improving efficiency of SegGen. The ideas behind the implemented improvements is to tune parameters of the algorithm during in its running. The first kind of improvements consists in the modification of the parameters and operators of the genetic algorithm used by SegGen along with the increasing quality of the generated population through the generations. The other improvements that have also been considered with the increasing in quality of the population as the process evolves is the taking into account of the nature of the coding of individuals: in this case individuals are segmentation instances, represented by binary vectors corresponding to the positions of the boundaries of the segmentations.

In this study, different types of the algorithm are performed on several test corpora. These types of experiment are various numbers of boundaries, fixed numbers of boundaries and stability test. Even though, the parameters of the algorithm in first results rest upon empirical values, first results are promising perspectives. Mixed type genetic operator tuning types give better results than the basic algorithm. Weighted type of the algorithm needs some improvement, because the improved implementation

of the algorithm can provide a qualified operation that assigning value to sentences regarding with their positions in the text.

REFERENCES

Arslanoglu, Y. (2006). Genetic Algorithm for Personnel Assignment Problem with Multiple Objectives. Master thesis, Middle East Technical University. Ankara, Turkey.

Callan, J.P. (1994). Passage-level evidence in document retrieval. In Bruce W. Croft and Cornelius J. Van Rijsbergen, editors, Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, Spring-Verlag, p.302 – 310.

Choi, F.Y.Y. (2000). Advances in domain independent linear text segmentation. Proc. of the ACL, Morgan Kaufmann Publishers Inc., pp. 26-33.

Coppin, B. (2004). Artificial Intelligence Illuminated, 1st ed., Sudbury MA: Jones and Bartlett Publishers.

Gaspar, P.M.S. (2010). Gene optimization for heterologous expression. Msc Thesis, Universidade de Aveiro, Portuguese.

Hamadi, Y., Monfroy, E. and Saubion, F. (2008). What is Autonomous Search?, TechReport MSR-TR-2008-80, Microsoft Research.

Harman, D. (1993). Overview of the first trec conference. In SIGIR'*93*, ACM Press, p. 36-47.

Hearst, M. A., Plaunt, C. (1993). Subtopic structuring for full-length document access. In Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Association for Computing Machinery, p.56–68.

Hearst, M.A. (1997). Texttilling: segmenting text into multi-paragraph subtopic passages. Comp. Ling., vol. 23(1), p.33-64.

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems, University of Michigan Press, USA.

Kan, M., Klavans, J. and McKeown, K. (1998). Linear segmentation and segment significance. 6th Workshop on Very Large Corpora (WVLC- 98), ACL SIGDAT, p.197-205.

Lamprier, S., Amghar, T., Levrat, B. and Saubion, F. (2007a). On evoluation methodologies for text segmentation algorithms. In Proc. of ICTAI'07 Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, vol. 2, p.19-26.

Lamprier, S., Amghar, T., Levrat, B. and Saubion, F. (2007b). SegGen: a genetic algorithm for linear text segmentation. In Proc. of the 20th International Joint conference on Artifical Intelligence, p.1647- 1652.

Lamprier, S., Amghar, T., Levrat, B. and Saubion, F. (2007c). ClassStruggle: a clustering based text segmentation. In Proc. of the 2007 ACM symposium on Applied computing, p.600-604.

Lamprier, S., Amghar, T., Levrat, B. and Saubion, F. (2008). Toward a more global and coherent segmentation of texts. Applied Artificial Intelligence, vol. 22(3), p.208-234.

Manning, C.D. and Schutze, H. (1999). Foundations of Statistical Natural Language Processing, MIT Press Cambridge, MA, USA.

Manning, C.D., Raghavan, P. and Schutze, H. (2008). Introduction to Information Retrieval, Cambridge University Press, UK.

McDonald, D., Chen, H. (2002). Using sentence-selection heuristics to rank text segments in txtractor. JCDL'02, ACM Press, p.28-35.

Mitchell, M. (1999). An Introduction to Genetic Algorithms, A Bradford Book The MIT Press, London, England.

Mochizuki, M., Honda, T., Okumura, M. (1998). Text Segmentation with Multiple Surface Linguistic Cues. COLING-ACL, p.881-885.

Pevzner, L. and Hearst, M.A. (2002). A critique and improvement of an evaluation metric for text segmentation. Comp. Ling., vol. 28(1), p.19-36.

Porter, M.F. (1980). An algorithm for suffix stripping, Program, 14(3) p.130−137.

Reynar, J.C. (2000). Topic Segmentation: Algorithms and Applications. Phd thesis, University of Pennsylvania, Seattle, WA.

Salton, G., Singhal, A., Buckley, C. and Mitra, M. (1996). Automatic text decomposition using text segments and text themes. Hypertext'96, ACM, p.53-56.

Slidewiki. (2013). Cosine Similarity Illustrated. URL: http://slidewiki.org/upload/media/images/29/656.png. [accessed May, 2013].

Wikipedia. (2013). Pareto Efficiency. URL: http://en.wikipedia.org/wiki/File:Front_pareto.svg. [accessed May, 2013].

Wikipedia. (2013). Fitness proportionate selection. URL: http://upload.wikimedia.org/wikipedia/commons/2/2a/Fitness_proportionate_selection_example.png. [accessed May, 2013].

Wikipedia. (2013). Crossover. URL: http://upload.wikimedia.org/wikipedia/en/b/b4/SinglePointCrossover.png. [accessed May, 2013].

Woolf, V. (2003). Orlando. Wordsworth Editions, London, UK.

Zitzler, E. (1999). Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Phd thesis, Swiss Federal Institute of Technology Zurich. Zurich, Switzerland.

BIOGRAPHICAL SKETCH

Neslihan Şirin Saygılı was born in 1985 in Gaziantep, Turkey. She finished Gaziantep Vehbi Dinçerler Science High School in 2004 and received her Bachelor's degree of Computer Engineering in 2011 from the Istanbul Technical University in Istanbul, Turkey. She is currently pursuing a Master's degree in Computer Engineering at Galatasaray University.

Her first publication is a workshop paper written under the supervision of the Prof. Dr. Bernard Levrat and Assoc. Prof. Dr. Tankut Acarman. Its title is "Managing Genetic Algorithm Parameters to Improve SegGen, a Thematic Segmentation Algorithm" has been accepted for publication in the IEEE proceedings of the 10th International Workshop on Text-based Information Retrieval. She has written an extended and improved version of her first paper, entitled "Tuning the Parameters of Genetic Algorithm for a Thematic Segmentation", which was submitted to IEEE International Conference on Tools with Artificial Intelligence and is currently under review.