Search Engine Implementation Using Evolutionary Technique

Enrollment number - 101215

Name of Student - Nitish Singla

Name of Supervisor - Dr. Pardeep Kumar

May – 2014

Submitted in partial fulfillment of the Degree of

Bachelor of Technology

in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

TABLE OF CONTENT

CERTIFICATE

This is to certify that the work titled **Search engine implementation using evolutionary technique** submitted by **Nitish Singla** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science Engineering to Jaypee University of Information Technology, Waknaghat, has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor: ……

Name of Supervisor : Dr. Pardeep Kumar

Designation : Assistant Professor (Senior Grade)

Date : ……

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and deep regards to my guide **Dr. Pardeep Kumar** for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

We feel motivated and encouraged every time we attend his meeting. Without his encouragement and guidance this project would not have materialized.

The guidance and support received from all the members who contributed to this project, was vital for the success of this project. We are grateful for their constant support and help.

Signature of the student : ……..

Name of Student : ……..

Date : …….

ABSTRACT

In this project, I presented a prototype of a search engine which makes heavy use of the structure present in hypertext. It is designed to crawl and index the given documents efficiently and produce satisfying search results. To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens to millions of queries every day. Despite the importance of large-scale search engines on the web, very little academics research has been done on them. Furthermore, due to rapid advance in technology and web proliferation, creating a web search engine today is very different from four years ago. This report provides an in-depth description of our web search engine. Apart from the problems of scaling traditional search techniques to data of this magnitude, there are new unique technical challenges involved with using the additional information present in hypertext to produce better search results. This report provides you with the details of how to build a practical system which can exploit the additional information present in hypertext.

CHAPTER 1.INTRODUCTION

Search engine is a program that search documents for specified documents and returns a list of the documents where the keywords were found. It is actually a general class of programs, however, the term is often used to specifically describe systems like Google, Bing and Yahoo! Search that enable users to search for documents on the World Wide Web.

1.1. History

During early days there was a list of web servers by Tim Berners-Lee and hosted on the CERN web server. One historical snapshot of this list in 1992 remains, but as more web servers went online the central list could no longer keep up. On the NCSA site, new servers were announced under the title "What's New!".The first tool used for searching on the Internet was Archie. It was created in 1990 by Alan Emtage, Bill Heelan and J. Peter Deutsch, computer science students at McGill University in Montreal. It downloaded the directory listings of all the files located on public anonymous FTP (File Transfer Protocol) sites, creating searchable database of file names; however, Archie did not index the contents of these sites since the amount of data was so limited it could be readily searched manually. Soon, many search engines came into light. These included Magellan, Excite, Infoseek, Inktomi, Northern Light, and AltaVista. Yahoo! was among the most popular ways for people to find web pages of interest, but its search function operated on its web directory, rather than its full-text copies of web pages. Information seekers could also browse the directory instead of doing a keyword-based search.Google adopted the idea of selling search terms in 1998, from a small search engine company named *goto.com*. This move had a significant effect on the SE business, which went from struggling to one of the most profitable businesses in the internet.Search engines were also known as some of the brightest stars in the Internet investing frenzy that occurred in the late 1990s. Several companies entered the market spectacularly, receiving record gains during their initial public offerings. Some have taken down their public search engine, and are marketing enterprise-only editions, such as Northern Light. Many search engine companies were caught up in the dot-com bubble, a speculation-driven market boom that peaked in 1999 and ended in 2001.By 2000, Yahoo! was providing search services based on

Inktomi's search engine. Yahoo! acquired Inktomi in 2002, and Overture (which owned All the Web and AltaVista) in 2003. Yahoo! switched to Google's search engine until 2004, when it launched its own search engine based on the 7combined technologies of its acquisitions.Microsoft first launched MSN Search in the fall of 1998 using search results from Inktomi. In early 1999 the site began to display listings from Looksmart, blended with results from Inktomi. For a short time in 1999, MSN Search used results from AltaVista were instead. In 2004, Microsoft began a transition to its own search technology, powered by its own web crawler (called msnbot).Microsoft's rebranded search engine, Bing, was launched on June 1, 2009. On July 29, 2009, Yahoo! and Microsoft finalized a deal in whichYahoo! Search would be powered by Microsoft Bing technology.

1.2. Need of Search Engine

Internet is the most popular medium to communicate with the visitors, customers and other businessmen for branding and promoting a business. Internet has changed the lives of all people of the world. Now everyone is depending on the internet to seek anything they need. It has become a very important part of our life. Every day the number of internet users is increasing due to their own reasons. In such a world it becomes necessary for every business to market itself in the online world, as it is a popular medium that is used by almost everyone in the world. A business can get benefit by using internet as a medium to market the products and services it is offering Therefore all businesses are making efforts to build a large customer base by means of internet. For getting large customer base it is necessary to be indexed in the search engines like Google, Yahoo, Bing and MSN etc. as internet users search any information they need through search engines. So, for getting indexed in search engines, it is needed to make efforts like SEO that stands for Search Engine Optimization. Search Engine Optimization is a technique to improve the online visibility of a business by making its website to be appeared in the top results of search engines.

1.3.Current Scenerio

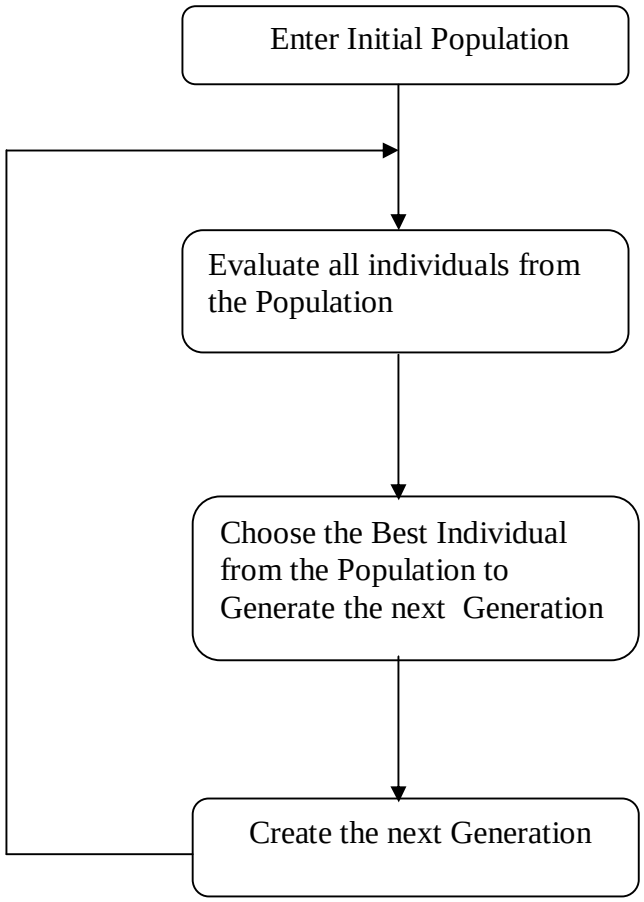
Due to the advent of e-commerce and corporate internets lots of data are available on the internet. This has led to the growth of organizational repositories containing large, complex, unstructured and fragmented data in the form of documents. Information retrieval systems are designed for storing, maintaining and searching large-scale sets of unstructured documents on the internet. But the web creates new challenges for information retrieval. The amount of information on the web is growing rapidly, as well as the number of new users inexperienced in the art of web research.

Search engine technology has had to scale dramatically to keep up with the growth of the web. In 1994, one of the first web search engine, the World Wide Web Worm(WWWW) had an index of 110,000 web pages and web accessible documents. As of November 1997, the top search engines claim to index from 2 millions to 100 millions web documents. By the year 2000, a comprehensive index of the Web contained over a billion documents. At the same time, the number of queries search engines handle has grown incredibly too. In March and April 1994, the World Wide Web Worm received an average of about 1500 queries per day. So today ,in search engine fast crawling technology is needed that can gather the web documents and keep them up to the date. Storage space must also be used efficiently to store indices and optionally, the documents themselves.For this, Finetuning the performance of information retrieval systems is essential. One step in optimizing the information retrieval experience is the deployment of Evolutionary Algorithms.

CHAPTER 2: EVOLUTIONARY TECHNIQUES

Evolutionary Techniques are stochastic search methods that mimic the metaphor of natural biological evolution. Evolutionary algorithms operate on a population of potential solutions applying the principle of survival of the fittest to produce better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

Flowchart



In the first step, we generate a Initial set of Population called Documents on which we want to apply search method. Then we entered in Evolutionary Algorithm Generational Loop. In the second step, we evaluate all the documents that are in our initial population and found ones that are relevant to user's query. In the third step, we evaluate the fitness of all the documents that are retrieved in second step. Then we select two documents with highest fitness value as Parents to find next generation. Now with the help of these parents decendents are reproduced to find next better generation. This process continues until we get the documents that best define the user's query.

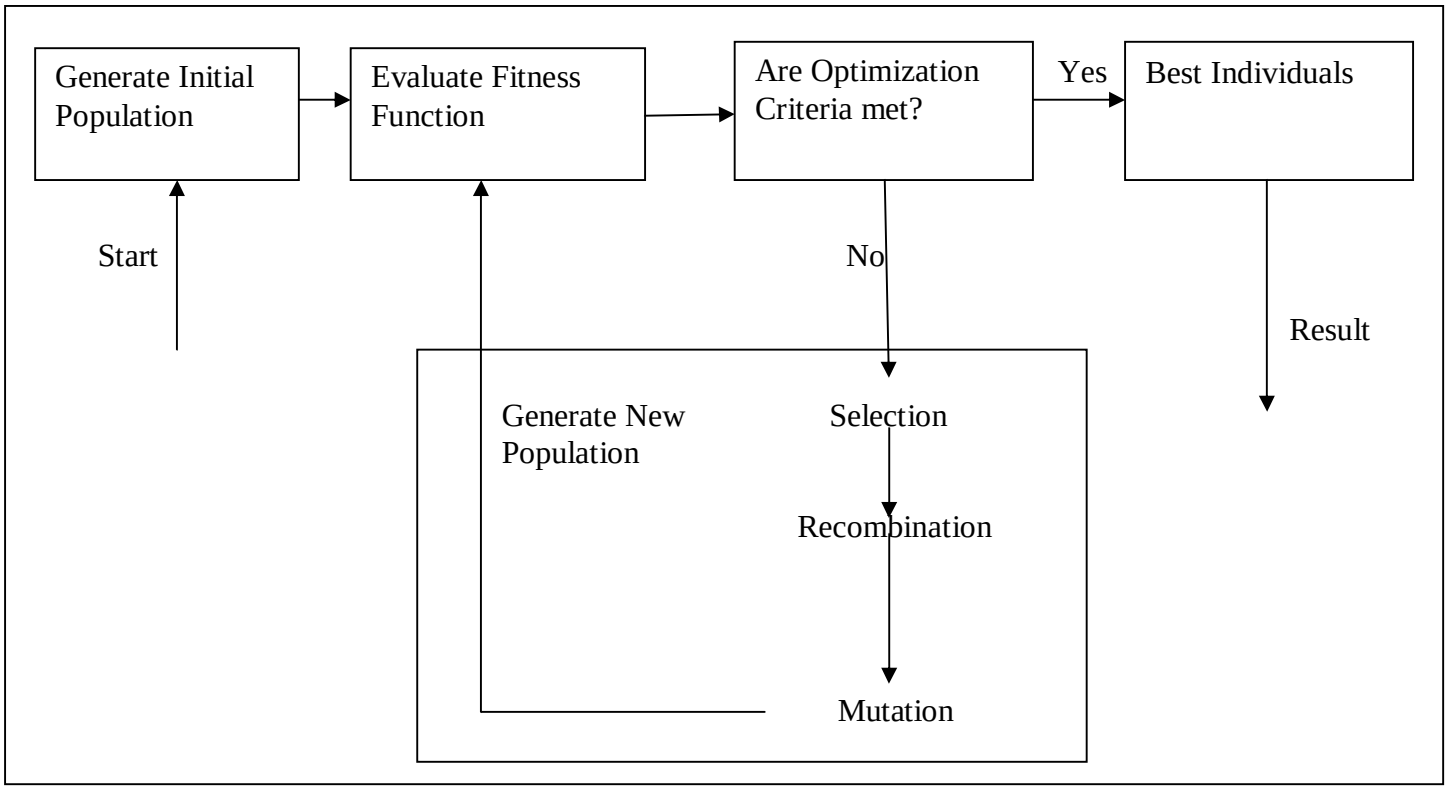
In the late sixties, J.H.Holland from the university of Michigam introduced the concept of sexual reproduction to Evolutionary Algorithm. Since then Holland is believed to be the father of the scientific field of Evolutionary Algorithm.Evolutionary Algorithm can be further separated into several more sub categories. One of which is Genetic Algorithm.

2.1 Genetic Algorithm

In the computer science field of artificial intelligence, genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic (also sometimes called a meta heuristic) is routinely used to generate useful solutions to optimization and search problems.[1] Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

Genetic Algorithms exhibit the clearest mapping from the natural process of evolution onto a computer system, because they stress the coding of attributes into a set of genes. One very common coding of attributes is a binary coding into a Bit String representing the genes. In contrast to Genetic Algorithms the Evolutionary Strategies stress the actual expression of an attribute and omit any redundant coding. This way they are able to move very efficiently through real valued search spaces by the use of specialized mutation operators. Genetic Programming (GP) is related to Genetic Algorithms regarding the general processing scheme. But instead of representing attributes in a general binary coding, Genetic Programming is specialized on representing programs or instruction sets as attributes. And the structures of these programs are altered during the optimization process.

Figure below shows the structure of a simple Genetic algorithm.



At the beginning of the computation a number of individuals (the population) are randomly initialized. The objective function is then evaluated for these individuals. The first/initial generation is produced. If the optimization criteria are not met the creation of a new generation starts. Individuals are selected according to their fitness for the production of offspring. Parents are recombined to produce offspring. All offspring will be mutated with a certain probability. The fitness of the offspring is then computed. The offspring are inserted into the population replacing the parents, producing a new generation. This cycle is performed until the optimization criteria are reached.

The basic idea is that if only those individuals of a population reproduce, which meet a certain selection criteria, and the other individuals of the population die, the population will converge to those individuals that best meet the selection criteria.

Algorithm

Given a clearly defined problem to be solved and a bit string representation for candidate solutions, a simple

GA works as follows:

1. Start with a randomly generated population of n l−bit chromosomes (candidate solutions to a problem).

2. Calculate the fitness $f(x)$ of each chromosome x in the population.

3. Repeat the following steps until n offspring have been created:

a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done "with replacement," meaning that the same chromosome can be selected more than once to become a parent.

b. With probability pc (the "crossover probability" or "crossover rate"), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no

crossover takes place, form two offspring that are exact copies of their respective parents.

(Note that here the crossover rate is defined to be the probability that two parents will cross

over in a single point. There are also "multi−point crossover" versions of the GA in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.)

c. Mutate the two offspring at each locus with probability pm (the mutation probability or

mutation rate), and place the resulting chromosomes in the new population. If n is odd, one new population member can be discarded at random.

4. Replace the current population with the new population.

5. Go to step 2

Components of Genetic Algorithm

- Representation
- Evaluation Function(Fitness Function)
- Population
- Parent Selection Mechanism
- GA Individual
- Mutation
- Crossover

Representation

Genetic representation is a way of representing individuals in evolutionary computation methods.It use linear binary representations. The most standard one is an array of bits. Arrays of other types and structures can be used in essentially the same way.

The first step in GA is to link the real world to GA world that to create a bridge between the original problem and the problem solving space where evolution will takes place .Objects forming possible solutions within the original problem context are referred to as phenotypes and their encoding,the individuals or documents in the GA are called Genotypes. The first design step is commonly called Representation,as it amounts to specifying a mapping from the phenotypes onto a set of Genotypes that are said to represent these Phenotypes.

Example:-Suppose the set of integers would form the set of Phenotypes. Now we will decide to represent them with their binary code.Hence 18 would be seen as a Phenotype and 10010 as Genotype representing it.

It is important to understand that the Phenotype space is very different from Genotype space and the whole evolutionary search takes place in the Genotype space.

Evaluation Function

The role of the Evolution Function is to represent the reqiurements to adapt to.It forms the basis for selection thereby it facilitates improvements.It gives an intuition about how good the individual is.It epends on directly to the problem. It is a function that assigns a quality measure to Genotypes.

Example:-If we were to maximize x^2 on the integers,the fitness of the genotype 10010 could be defined as the square of its corresponding Phenotype:18^2=324.

Evaluation Function is also called as Fitness Function.

Population

The major questions to consider are firstly the size of the population, and secondly

the method by which the individuals are chosen.It is the number of individuals on which we are using Evolutionary Techniques.Individuals are static objects not changing or adapting.It is initialized randomly. The size of the population has been approached from several theoretical points of view, although the underlying idea is always of a trade-off between efficiency and effectiveness. Intuitively, it would seem that there should be some 'optimal' value for a given string length, on the grounds that too small a population would not allow sufficient room for exploring the search space effectively, while too large a population would so impair the efficiency of the method that no solution could be expected in a reasonable amount of time.

Selection

The Genetic Algorithm performs a selection process in which the 'most-fit' members of the population survive, and the 'least-fit' members are eliminated. The process is the step that guides the GA towards even-better solutions. In Selection, we select two document as parents from our database. The Selection is directly proportional to the fitness value of the documents.

Most popular Selection Algorithms for GA are:

1. Roulette wheel selection
2. Tournament selection.

1. Roulette wheel selection

In this selection method, we list all the documents that contains data that is relevant to user's query.ents Then we add the fitness values of all these documents and called it T. Then we will generate a random number between 1 and T.Then we generate a random number for each document and add this value to document's fitness value. Now we select two documents whose fitness values are greater.

Example:-

Documents	1	2	3	4	5	6
Fitness	7	2	16	7	4	11
Running Total	8	10	27	34	38	49
N(1<N<50)	22	23	23	23	23	23
Selected			3			

So,in above example third document is selected

2. Tournament selection.

1. Binary tournament

I n this two documents are randomly selected according to their fitness value.

2.Larger tournaments

n individuals are randomly chosen and two most fit documents are chosen as parents

GA Individual

GA individuals store the solution attributes not directly in clear type but in a coded representation.Most common is the binary coding of an attribute in a chain of Bits, a Bit-String8.In,the fitness value of each individual is represented in Binary and now this binary value represents the individual.

Ex Suppose we have two individuals with fitness value 37 and 43.Now our individuals are:-

1. 0 0 1 0 0 1 0 1

2. 0 0 1 0 1 0 1 1

Mutation

The EA periodically makes random alterations in one or more members of the current population ,yielding a new candidate solution which may be better than the existing ones.It can be done by switching the bits or by updating the values**.**

Example Like in previous example, suppose we have two individuals

1. 0 0 1 0 0 1 0 1 **2.** 0 0 1 0 1 0 1 1

Now we apply mutation on these individuals by randomly changing their one bit with another.The result is:-

1. 0 0 1 0 0 1 1 0 **2.** 0 0 1 0 1 1 0 1

Crossover

The main distinguishing feature of a GA is the use of crossover.Cross-over method is to replace a gene of the individual with that of another individual.These individuals are the two parents that are selected in parent selection.

It is of two types:-

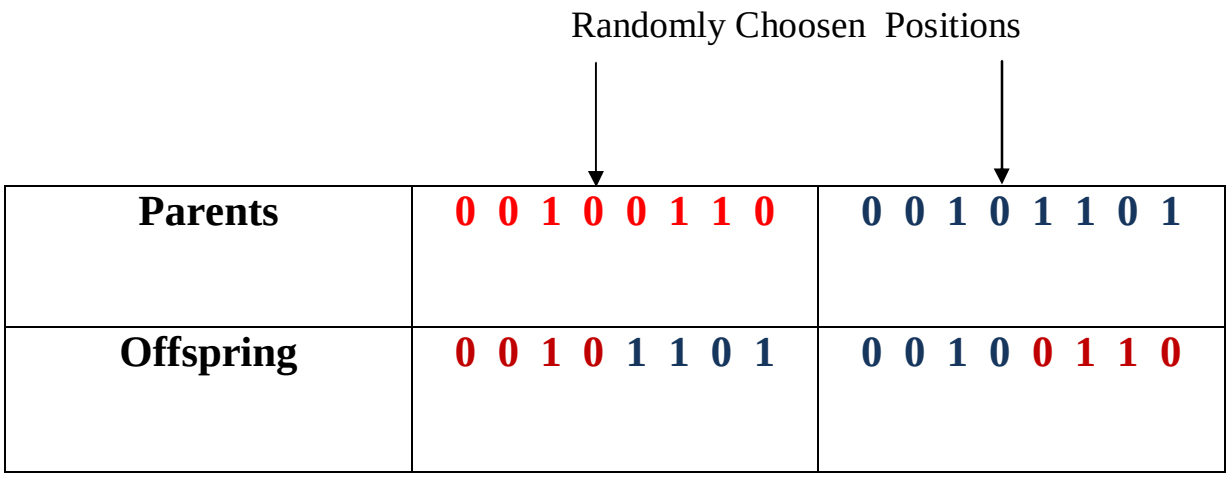
1. Single Point Crossover
2. Two point Crossover

1. Single Point Crossover

A single crossover position is chosen at random and the parts of two parents after the crossover position are exchanged to form two offspring. The idea here is, of course, to recombine building blocks (schemas) on different strings. Single−point crossover has some shortcomings, though. For one thing, it cannot combine all possible schemas. In this,we randomly choose one position in individuals and divide them into two parts.Now the first offspring is head of parent 1 with tail of parent 2 and the second offspring is head of parent 2 with tail of parent 1.

Figure below shows the Example of Single Point Crossover with two GA Individuals

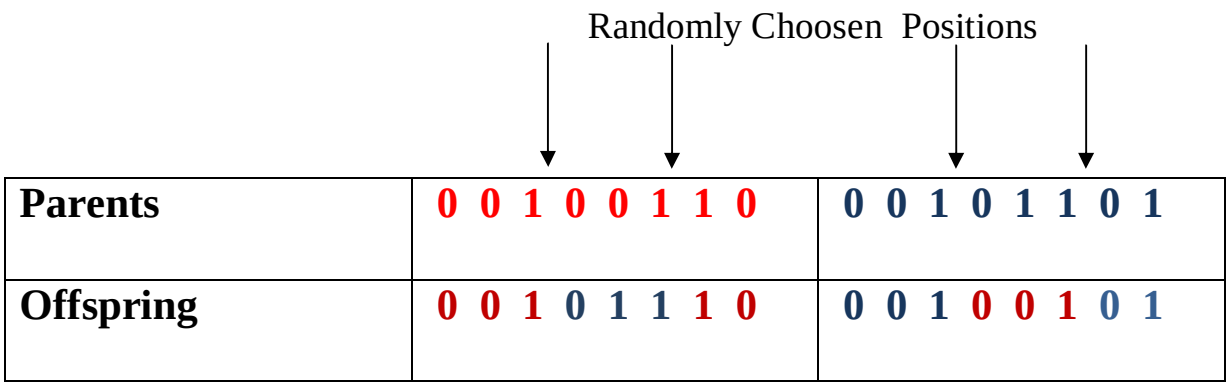
00100110 and 00101101



2.Two point crossover

In Two Point Crossover two positions are chosen at random and the segments between them are exchanged. Two−point crossover is less likely to disrupt schemas with large defining lengths and can combine more schemas than single−point crossover. In addition, the segments that are exchanged do not necessarily contain the endpoints of the strings.

Figure below shows the example of Two point Crossover



CHAPTER 3. REQUIREMENTS

Software Requirements

1. JDK 1.7
   - Eclipse

2. Database
   - Xampp
   - MySQL Database

3. Jdbc Driver for MySQL Database Server
   - mysql-connector-java-5.1.7-bin.jar

4. Operating System
   - Windows Vista / XP /7/8

Hardware Requirements:

1. Intel P4 processor with minimum 2.0Ghz Speed

2. RAM: Minimum 512MB

3. Hard Disk: Minimum 20GB

3.1 SYSTEM DESIGN

3.1.1 Tools and Technology used:

3.1.2 JAVA:

Java is a computer programming language that is concurrent, class-based, object-oriented, and designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere", meaning that code that runs on one platform does not need to recompile to run on another. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of the computer architecture. Java is, one of the most popular programming languages in use, particularly for the client-server web applications. Java was originally developed by James Gosling at Sun Microsystems and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has some facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were developed by Sun from 1991. As of May 2007, in compliance with the specifications of the Java Community Process, Sun licensed again most of its Java technologies under the General Public License. Others have also developed some implementations of these Sun technologies, such as the GNU Compiler for Java (byte code compiler), GNU Class path (standard libraries), and Iced Tea-Web (browser plug in for applets).

3.1.3 Swings (JAVA):

Swing is the primary Java GUI widget toolkit. It is part of Oracle's Java Foundation Classes — API to provide a graphical user interface for the java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier Abstract Window Toolkit . Swing provides a native look and feel that emulates the look of several platforms, and also supports a pluggable look and feel that allows applications to have a look unrelated to the underlying platform. It has more powerful and

flexible components than Abstract Window Toolkit. In addition to familiar components such as buttons, check boxes , Swing provides some advanced components such as tabbed panel, scroll panes, tables, and lists.

Unlike AWT components, Swing components are not implemented by platform-specific code. They are written entirely in Java and therefore are platform-independent. The term "lightweight" is used to describe such element.

3.1.4 Eclipse

The Eclipse Project is an open source project of eclipse.org, overseen by a Project Management Committee (PMC) and project leaders. The work is done in subprojects working against Git repositories. The Eclipse Project Charter describes the organization of the project, roles and responsibilities of the participants, and top level development process for the project. The JDT and PDE are plug-in tools for the Eclipse Platform. Together, these three pieces form the Eclipse SDK download, a complete development environment for Eclipse-based tools, and for developing Eclipse itself.

- **Eclipse Project Development**
  Release plans and other information about the Eclipse Project development process.
- **Downloads**
  Download the Eclipse SDK, Eclipse RCP, SWT, the Eclipse Java compiler, and many more. You can find the current release here. Or, download the latest stable and integration builds if you want to try out the newest features under development, or get started with contributing to the project.
- **Documentation**
  Browse the documentation included with Eclipse Project releases.

.

Eclipse Platform

The Platform defines the set of frameworks and common services that collectively make up "integration-ware" required to support the use of Eclipse as a component model, as a rich client platform (RCP) and as a comprehensive tool integration platform. These services and frameworks include a standard workbench user interface model and portable native widget

toolkit, a project model for managing resources, automatic resource delta management for incremental compilers and builders, language-independent debug infrastructure, and infrastructure for distributed multi-user versioned resource management.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their app. Among the features of the platform are:

- User interface management.
- User settings management.
- Storage management.
- Window management.
- Wizard framework.

Eclipse is a free, open-source, cross-platform IDE with built in support for java language.

3.1.5 MySQL

MySQL is the world's second most used open-source relational database management system . It is named after co-founder Michael Widenius's daughter(My). The SQL phrase stands for Structured Query Language.

The default port of Mysql is 3306. The MySQL development project has made its source code available under the terms of the General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by the single firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack. LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require full-featured database management system often use MySQL.

MySQL is a relational database management system , and ships with no graphic user interface tools to administer MySQL databases or manage data contained within the databases. Users may use the included command line tools, or use front-ends, desktop software and web applications that create and manage MySQL databases, build database

structures, back up data and work with data records. The MySQL front-end tools, MySQL Workbench is actively developed by Oracle, and is freely available for use.

CHAPTER 4: INDEXER

Search engine indexing collects, parses, and stores data to facilitate fast and accurate information retrieval. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan document in the corpus, which would require considerable time and computing power. For example, while an index of 10,000 documents can be queried within milliseconds, a sequential scan of every word in 10,000 large documents could take hours.

Any parser which is designed to run on the entire Web must handle a huge array of possible of a tag, non-ASCII characters, HTML tags nested hundreds deep, and a great variety of other errors the challenge anyone's imagination to come up with equally creative ones. For maximum speed, instead of using YACC to generate a CFG parser, we use flex to generate a lexical analyzers which we outfit with its own stack. Developing this parser which runs at a reasonable speed and is very robust involved a fair amount of work. Indexing Documents into Barrel. After each document is parsed, it is encoded into a number of barrels. Every word is converted into a worded by using an in-memory hash table-the lexicon. New additions to the lexicon hash table are logged to a file. Once the words are converted into wordID's , their occurrences in the current document are translated into hit lists and are written into the forward barrels. The main difficulty with parallelization of the indexing phase is that the lexicon needs to be shared. Instead of sharing the lexicon, we took the approach of writing a log of all the extra words that were not in a base lexicon, which we fixed at 14 million words. That way multiple indexers can run in parallel and then the small log file of extra word can be processed by one final indexer.

In order to generate the inverted index, the sorter takes each of the forward barrels and sorts it by wordID to produce an inverted barrel for title and anchor hits and a full text inverted barrel. This process happens one barrel at a time, thus requiring little temporary storage.

Also, we parallelize the sorting phase to use as many machines as we have simply running multiple sorters, which can process different buckets at the same time. Since the barrels don't fit into main memory, the sorter further subdivides them into baskets which do fit into main memory, the sorter further subdivides them into baskets which do fit into memory

based on worded and docID. Then, the sorter loads each basket into memory, sorts it and writes its contents into short inverted barrel and the full inverted barrel.

4.1. Merge Factors

How data enters the index, or how word or subjects features are added to the index during text corpus traversal, and whether multiple indexers can work asynchronously. The indexer must first check whether it is updating old content or adding new content.

4.2. Storage Techniques

How to store the index data, that is, whether information should be data compressed or filtered.

4.3. Index Size

How much computer storage is required to support the index.

4.4. Lookup Speed

How quickly a word can be found in the inverted index.

4.5. Maintenance

How the index is maintained over time.

4.6. Fault Tolerance

How important it is for the service to be reliable.

CHAPTER 5: SOFTWARE ENGINEERING ASPECTS OF PROJECT

Software Engineering is the application of a systematic, disciplined, quantifiable approach to the design, development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software. It is inter-disciplinary in nature. It has emerged as a discipline very recently. Due to rapid growth of knowledge in this field, software professionals and academicians felt the need to have a consistent view of software engineering worldwide. To achieve this objective the IEEE Computer Society's Professional Practices Committee has published a guide to Software Engineering Body of Knowledge (SWEBOK) in 2004. The guide is based on a generally accepted portion of the Body of Knowledge. The material that is recognized as being within this discipline is organized into ten knowledge areas as follows:

- **Software requirements:** The elicitation, analysis, specification, and validation of requirements for software.
- **Software design:** The process of defining the architecture, components, interfaces, and other characteristics of a system or component. It is also defined as the result of that process.
- **Software construction:** The detailed creation of working, meaningful software through a combination of coding, verification, unit testing, integration testing, and debugging.
- **Software testing:** The dynamic verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite executions domain, against the expected behavior.
- **Software maintenance:** The totality of activities required to provide cost-effective support to software.

- **Software configuration management:** The identification of the configuration of a system at distinct points in time for the purpose of systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the system life cycle.
- **Software engineering management:** The application of management activities—planning, coordinating, measuring, monitoring, controlling, and reporting—to ensure that the development and maintenance of software is systematic, disciplined, and quantified.
- **Software engineering process:** The definition, implementation, assessment, measurement, management, change, and improvement of the software life cycle process itself.
- **Software engineering tools and methods**: The computer-based tools that are intended to assist the software life cycle processes, see Computer Aided Software Engineering, and the methods which impose structure on the software engineering activity with the goal of making the activity systematic and ultimately more likely to be successful.
- **Software quality**: The degree to which a set of inherent characteristics fulfills requirements.

Software engineering is concerned with both technical as well as managerial aspects of software development. There are four important core aspects of software development. These four aspects are: (1) Product, (2) Process, (3) People and (4) Project.

- **Product:** Software, as a product, has to perform certain specific functions required by users (customers). Determination of correct functional requirements and features of software to be produced is a very critical activity of software development. For this, various stakeholders and users of software are identified to elicit information for determining functional specification of software. Sometimes requirements of one class of users may conflict with those of another. Finalization of

functional specification is often a balancing act of satisfying requirements of different stakeholders within cost and time constraints.

- **Process:** It refers to methodologies to be followed for developing the software. It is the framework for establishment of a comprehensive plan and strategies for software development. The process specifies the policies, procedures, tools and techniques to be used for software development. A number of models are available. CMM (Capability Maturity Model) is a widely used standard model for software development process.

- **People:** Software development requires creativity and knowledge work. It is often difficult to specify the quantitative and qualitative measures of this work. Since fast technological developments are taking place in the field of computer science, updating of knowledge is a part of computer professionals' job. A number of people having diverse expertise are required to work together for developing any software. Since work of one individual affects the work of others, quality software is mostly developed through teamwork. Hence, developing motivation, morale and teamwork among people and upgrading their professional expertise are important aspects of software engineering.

- **Project:** As stated earlier, there is a requirement for great amounts of effort, time and money to design and develop any commercial software. A number of interrelated activities have to be performed in a planned schedule for completing the software within time constraints. Hence, development of any software can be considered as a project. Thus, project management aspects such as planning and monitoring of activities, schedule, resources and expenditure are important for software development.

CHAPTER 6.SOURCE CODE

1. GeneticAlgo.java

```
import java.util.Random;
import java.util.Scanner;
public class GeneticAlgo {
private int[][] freqMatrix = null;
private int[][] orginalFreqMatrix = null;
private int[] fitnessMatrix = null;
private float[] relationMatrix = null;
private Scanner scanner = null;
private int min = -1;
private int count = 0;
private int passCount = 0;
private int crossover = 1;
private int docIndex = -1;
public String runAlgo(int[][] freqMat, int crossover) {
        this.crossover = crossover;
        StringBuffer buffer = new StringBuffer();
        freqMatrix = new int[freqMat.length][freqMat[0].length];
        orginalFreqMatrix = new int[freqMat.length][freqMat[0].length];
        for (int i = 0; i < freqMat.length; i++)
            {
                for (int j = 0; j < freqMat[0].length; j++)
                    {
                        freqMatrix[i][j] = freqMat[i][j];
                        orginalFreqMatrix[i][j] = freqMat[i][j];
                    }

            }
        count = orginalFreqMatrix[0].length;
```

```
do
{
passCount++;
buffer.append("Pass " + passCount + ":\n\n");
for (int i = 0; i < freqMatrix[0].length; i++)
  {
    buffer.append("[");
    for (int j = 0; j < freqMatrix.length; j++)
      {buffer.append(freqMatrix[j][i] + ","}
buffer.append("]\n\n");
  }
populateFitnessMatrix();
count = 0;
for (int i = 0; i < freqMatrix[0].length; i++)
 {
     if (fitnessMatrix[i] > min)
       {
          count++;
       }
 }
if (count < 2)
{
 boolean flag = false;
 for (int i = 0; i < orginalFreqMatrix[0].length; i++)
   {
     int sum = 0;
 for (int j = 0; j < orginalFreqMatrix.length; j++)
   {
        sum += orginalFreqMatrix[j][i];
    }
if (sum > min)
```

```
                    {
                      buffer.append("Most relative doc is : Doc " + (i + 1));
                docIndex=i;
                flag = true;
                return buffer.toString();
                }
                }
                if (!flag)
                {
                int max = 0;
                for (int j = 0; j < orginalFreqMatrix.length; j++)
                 {
                max += orginalFreqMatrix[j][0];
                 }
                int docMax = 0;
                for (int i = 1; i < orginalFreqMatrix[0].length; ++i)
                    {
                      int localMax = 0;
                for (int j = 0; j < orginalFreqMatrix.length; j++)
                     {
                          localMax += orginalFreqMatrix[j][i] + orginalFreqMatrix[j][i];
                      }
                if (max < localMax) {
                max = localMax;
  docMax = i;
          }
          }
buffer.append("Most relative doc is : Doc " + (docMax + 1));
docIndex=docMax;
return buffer.toString();
          }
```

```
break;
} else if (count < freqMatrix[0].length)
        {
           int[][] freqMatrixLcl = new int[freqMatrix.length][count];
           int lc = 0;
for (int i = 0; i < freqMatrix[0].length; i++)
 {
    if (fitnessMatrix[i] > min)
      {
         for (int j = 0; j < freqMatrix.length; j++)
           {
              freqMatrixLcl[j][lc] = freqMatrix[j][i];
           }
         lc++;
         }
}
freqMatrix = new int[freqMatrix.length][count];
for (int i = 0; i < freqMatrixLcl[0].length; i++) {
for (int j = 0; j < freqMatrixLcl.length; j++)
  {
    freqMatrix[j][i] = freqMatrixLcl[j][i];
  }
}
populateFitnessMatrix();
}
populateRelationMatrix();
float[] relationMatrixLcl = sortRelationMatrix();
int[] val = new int[freqMatrix[0].length];
for (int i = 0; i < relationMatrix.length; ++i) {
for (int k = 0; k < fitnessMatrix.length; k++) {
if (relationMatrixLcl[i] == relationMatrix[k]) {
```

```
val[i] = fitnessMatrix[k];
}
}}
String st1 = mutation(val[0]);
String st2 = mutation(val[1]);
String[] arras = crossover(st1, st2);
int a = binaryToDecimal(arras[0]);
int b = binaryToDecimal(arras[1]);
if (a < b) {min = a;
}
else {
min = b;
}
} while (true);
return buffer.toString();
}
public int binaryToDecimal(String binary)
{
char[] charArray = binary.toCharArray();
int answer = 0;
int count = 0;
for (int index = charArray.length - 1; index >= 0; index--)
 {
if (charArray[index] == '1')
{
answer = answer + (int) Math.pow(2, count);
}
count++;
}return answer;
}
public String[] crossover(String bin1, String bin2) {
```

```
if (crossover == 1)
 {
String bin3 = bin1.substring(0, 4) + bin2.substring(4);
String bin4 = bin2.substring(0, 4) + bin1.substring(4);
String[] strings = new String[2];
strings[0] = bin3;
strings[1] = bin4;
return strings;} else
{
String bin3 = bin1.substring(0, 2) + bin2.substring(2, 4)+ bin1.substring(4, 6) +
bin2.substring(6);
String bin4 = bin2.substring(0, 2) + bin1.substring(2, 4)+ bin2.substring(4, 6) +
bin1.substring(6);
String[] strings = new String[2];
strings[0] = bin3;
strings[1] = bin4;
return strings;}}public String mutation(int ch)
 {
String binary = printBinaryFormat(ch);
char[] charArr = binary.toCharArray();
Random ran = new Random();
int r = ran.nextInt(4);
char c = charArr[r + 4];
if (c == '1')
 {
charArr[r + 4] = '0';
} else {
charArr[r + 4] = '1';
}
r = ran.nextInt(4);
c = charArr[r + 4];
```

```
if (c == '1')
{
charArr[r + 4] = '0';} else
{
charArr[r + 4] = '1';
}
binary = new String(charArr);
return binary;
}
public String printBinaryFormat(int number)
 {
int binary[] = new int[8];
int index = 0;
String binaryString = "";
while (number > 0) {
binary[index++] = number % 2;
number = number / 2;
}
for (int i = 7; i >= 0; i--)
{
binaryString += binary[i];
}
return binaryString;}
public float[] sortRelationMatrix()
{
float[] relationMatrixLcl = new float[count];
relationMatrixLcl = relationMatrix.clone();
for (int i = 0; i < relationMatrixLcl.length; i++)
 {
for (int j = i + 1; j < relationMatrix.length; j++)
 {
```

```
if (relationMatrixLcl[i] < relationMatrixLcl[j])
{
float t = relationMatrixLcl[i];
relationMatrixLcl[i] = relationMatrixLcl[j];
relationMatrixLcl[j] = t;
}
}}
return relationMatrixLcl;}
public void populateRelationMatrix()
{
relationMatrix = new float[count];
for (int i = 0; i < freqMatrix[0].length; i++)
{
float relValue = 1;
for (int j = 0; j < freqMatrix.length; j++)
{
relValue *= freqMatrix[j][i];
}
for (int j = 0; j < freqMatrix.length; j++)
{
for (int k = j + 1; k < freqMatrix.length; k++)
{
relValue = relValue
/ Math.abs((freqMatrix[j][i] - freqMatrix[k][i]) == 0 ? 0.5f
(freqMatrix[j][i] - freqMatrix[k][i]));
}
}
relationMatrix[i] = relValue;}
        }private void populateFitnessMatrix()
{
fitnessMatrix = new int[count];
```

```
for (int i = 0; i < count; i++)
{
int sum = 0;
for (int j = 0; j < freqMatrix.length; j++)
{
sum += freqMatrix[j][i];
}
fitnessMatrix[i] = sum;
}
}
public int getDocIndex() {
return docIndex;}
public void setDocIndex(int docIndex) {
this.docIndex = docIndex;}}
```

2.MainUI.java

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Vector;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JOptionPane;
```

```
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTabbedPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;
public class MainUI extends JFrame implements ActionListener, ChangeListener{
private JTabbedPane tabbedPane = null;
private JPanel searchDocumentPanel = null;
private JPanel uploadDocumentPanel = null;
private JPanel viewDocumentsPanel = null;
private JTextField searchText = null;
private JList searchDocumentList = null;
private Vector<Option> searchDocumentListVector = null;
private JButton searchDocumentButton = null;
private JComboBox crossoverType = null;
private JTextArea searchOutputText = null;
private JTextField documentHeader = null;
private JTextArea documentText = null;
private JButton documentButton = null;
private JButton editDocumentButton = null;
private JButton deleteDocumentButton = null;
private JList documentList = null;
private Vector<Option> documentListVector = null;
private Long docId=null;
private int docIndex=-1;
Long[] docIds=null;
private JButton openDocumentButton = null;
private Connection connection=null;
public MainUI() {
```

```
makeConnection();initUI();}private void makeConnection() {try {
Class.forName("com.mysql.jdbc.Driver");
connection  = DriverManager.getConnection("jdbc:mysql://localhost/genetic_algo", "root",
"root");
}catch (Exception e)
{e.printStackTrace();}}private void initUI()
 {
tabbedPane = new JTabbedPane();
initUploadDocumentPanel();
this.add(tabbedPane);
Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
double width = screenSize.getWidth();
double height = screenSize.getHeight();
this.setSize((int)width, (int)height);
this.setVisible(true);}
private void initUploadDocumentPanel()
 {
searchDocumentPanel = new JPanel(null);
uploadDocumentPanel = new JPanel(null);
viewDocumentsPanel = new JPanel(null);
JLabel searchTextLabel = new JLabel("Enter search text");
searchText = new JTextField();
searchDocumentListVector = new Vector<Option>();
searchDocumentList=new JList();
documentListVector = new Vector<Option>();
documentListVector=listDocuments();
searchDocumentList.setListData(documentListVector);
Vector<Option> crossoverTypeListVector = new Vector<Option>();
crossoverTypeListVector.add(new Option("SINGLE_POINT", "Single Point"));
crossoverTypeListVector.add(new Option("TWO_POINT", "Two Point"));
JLabel crossoverTypeLabel = new JLabel("Crossover Type");
```

```
crossoverType = new JComboBox(crossoverTypeListVector);
searchDocumentButton = new JButton("Search");
searchDocumentButton.addActionListener(this);
openDocumentButton=new JButton("Open Document");
openDocumentButton.addActionListener(this);
searchOutputText = new JTextArea();
                JScrollPane sp3 = new JScrollPane(searchOutputText);

JScrollPane sp2 = new JScrollPane(searchDocumentList);
searchTextLabel.setBounds(200, 20, 150, 25);
searchText.setBounds(330, 20, 570, 25);
sp2.setBounds(200, 60, 700, 120);
crossoverTypeLabel.setBounds(200, 200, 100, 25);
crossoverType.setBounds(320, 200, 150, 25);
searchDocumentButton.setBounds(320, 240, 100, 30);
sp3.setBounds(200, 300, 700, 300);
openDocumentButton.setBounds(340, 620, 160,
30);searchDocumentPanel.add(searchTextLabel);
searchDocumentPanel.add(searchText);
searchDocumentPanel.add(sp2);
searchDocumentPanel.add(crossoverTypeLabel);
searchDocumentPanel.add(crossoverType);
searchDocumentPanel.add(searchDocumentButton);
searchDocumentPanel.add(openDocumentButton);
searchDocumentPanel.add(sp3);
JLabel uploadDocHeadLabel = new JLabel("Enter document heading");
documentHeader = new JTextField();
JLabel uploadDocLabel = new JLabel("Enter text of document to be uploaded");
documentText = new JTextArea(50, 20);
documentButton = new JButton("Submit");
documentButton.addActionListener(this);
```

```
JLabel documentListLabel = new JLabel("List of available documents");
documentList=new JList();
JScrollPane sp1 = new JScrollPane(documentList);
editDocumentButton = new JButton("Edit");
editDocumentButton.addActionListener(this);
deleteDocumentButton = new JButton("Delete");
deleteDocumentButton.addActionListener(this);
documentListLabel.setBounds(500, 65, 250, 25);
sp1.setBounds(250, 100, 700, 250);
editDocumentButton.setBounds(450, 380, 100, 30);
deleteDocumentButton.setBounds(630, 380, 100, 30);
viewDocumentsPanel.add(documentListLabel);
viewDocumentsPanel.add(sp1);
viewDocumentsPanel.add(editDocumentButton);
viewDocumentsPanel.add(deleteDocumentButton);
uploadDocHeadLabel.setBounds(540, 10, 300, 25);
documentHeader.setBounds(430, 40, 380, 25);
uploadDocLabel.setBounds(500, 70, 300, 25);
documentButton.setBounds(540, 655, 180, 30);
JScrollPane sp = new JScrollPane(documentText);
sp.setBounds(150, 100, 1000, 530);
uploadDocumentPanel.add(uploadDocHeadLabel);
uploadDocumentPanel.add(documentHeader);
uploadDocumentPanel.add(uploadDocLabel);
uploadDocumentPanel.add(sp);
uploadDocumentPanel.add(documentButton);
tabbedPane.addTab("Search Document", searchDocumentPanel);tabbedPane.addTab("Add
Document", uploadDocumentPanel);
tabbedPane.addTab("View Documents", viewDocumentsPanel);
tabbedPane.addChangeListener(this);
}
```

```
public static void main(String[] args)
{
new MainUI();}public void actionPerformed(ActionEvent e)
{
if (e.getSource() == documentButton)
{
String documentHeaderText = documentHeader.getText();
String documentDetailText = documentText.getText();
updateDocument(docId,documentHeaderText,documentDetailText);
}
else if(e.getSource()==editDocumentButton)
{
if(documentList.getSelectedValue()!=null)
{
Long id=Long.valueOf(((Option)documentList.getSelectedValue()).getKey());
editDocument(id);
}
}
else if(e.getSource()==deleteDocumentButton)
{
if(documentList.getSelectedValue()!=null)
{
Long id=Long.valueOf(((Option)documentList.getSelectedValue()).getKey());
deleteDocument(id);
documentListVector=listDocuments();
documentList.setListData(documentListVector);
this.validate();
}
}
else if(e.getSource()==searchDocumentButton)
{
```

```
String searchDocText=searchText.getText();
Object[] selectedDocs=searchDocumentList.getSelectedValues();
docIds=new Long[selectedDocs.length];
for (int i = 0; i < docIds.length; i++) {
{
docIds[i]=Long.parseLong(((Option)selectedDocs[i]).getKey());
}
String[] wordsArray=skipWord(searchDocText).split(" ");
String[] documentText=getDocTexts(docIds);
int[][] freqMatrix=new int[wordsArray.length][docIds.length];
for (int i = 0; i < wordsArray.length; i++)
 {
for (int j = 0; j < documentText.length; j++)
{
freqMatrix[i][j]=(documentText[j].length()-
documentText[j].replaceAll(wordsArray[i],"").length())/wordsArray[i].length();
}}
for (int i = 0; i < freqMatrix.length; i++)
{
for (int k = 0; k < freqMatrix[0].length; k++)
{
System.out.println(freqMatrix[i][k]);
}
System.out.println("\n\n");
}String usefulWord="[";
for (int i = 0; i < wordsArray.length; i++)
 {
usefulWord+=wordsArray[i]+",";}usefulWord+="]";
GeneticAlgo ga=new GeneticAlgo();
if(((Option)crossoverType.getSelectedItem()).getKey().equals("SINGLE_POINT"))
{
```

```
searchOutputText.setText("Useful Word :
"+usefulWord+"\n\n"+ga.runAlgo(freqMatrix,1));}else
{
searchOutputText.setText("Useful Word : "+usefulWord+"\n\n"+ga.runAlgo(freqMatrix,2));
}
this.docIndex=ga.getDocIndex();
}else if(e.getSource()==openDocumentButton)
{
if(docIndex!=-1)
{
editDocument(docIds[docIndex]);
tabbedPane.setSelectedIndex(1);}
}
private String[] getDocTexts(Long[] ids)
{
String[] arr=new String[ids.length];
try
QERW{
String selectSQL="select doc_text from documents where id in (";
for (int i = 0; i < ids.length; i++) {
selectSQL+=ids[i]+",";
}
selectSQL=selectSQL.substring(0,selectSQL.length()-1);
selectSQL+=")";
PreparedStatement pstmt=connection.prepareStatement(selectSQL);
ResultSet rs=pstmt.executeQuery();
int count=0;
while(rs.next())
{
arr[count]=rs.getString(1);
count++;
```

```
}
}
catch (Exception e) {e.printStackTrace();
}
return arr;
}
private String skipWord(String searchDocText)
{
String updated=" "+searchDocText;
try
{
String selectSQL="select word from skip_words";
PreparedStatement pstmt=connection.prepareStatement(selectSQL);
ResultSet rs=pstmt.executeQuery();
while(rs.next()){
updated=updated.replaceAll(" "+rs.getString(1),"");
}
} catch (Exception e)
{
e.printStackTrace();}
System.out.println(updated.trim()+"hellllll");return updated.trim();
        }
private void editDocument(Long id)
{
try
{
String selectSQL="select heading,doc_text from documents where id="+id;
PreparedStatement pstmt=connection.prepareStatement(selectSQL);
ResultSet rs=pstmt.executeQuery();
if(rs.next())
{
```

```
docId=id;
String heading=rs.getString(1);
String text=rs.getString(2);
documentHeader.setText(heading);
documentText.setText(text);
tabbedPane.setSelectedIndex(1);}} catch (Exception e)
{
e.printStackTrace();
}
}
private void updateDocument(Long id,String header, String text)
 {
try
{
String updateSQL=null;
if(docId==null){updateSQL="insert into documents values(null,'"+header+"','"+text+"')";
}else
{
updateSQL="update documents set heading='"+header+"',doc_text='"+text+"' where
id="+id;
}
PreparedStatement pstmt=connection.prepareStatement(updateSQL);
pstmt.executeUpdate();
docId=null;} catch (Exception e)
{
e.printStackTrace();}
documentHeader.setText("");
documentText.setText("");
docId=null;
JOptionPane.showMessageDialog(this,"Updated Successfuly");
}private void deleteDocument(Long id)
```

```
{
try {
String deleteSQL="delete from documents where id="+id;
PreparedStatement pstmt=connection.prepareStatement(deleteSQL);
pstmt.executeUpdate();} catch (Exception e)
{
e.printStackTrace();}}
private Vector<Option> listDocuments()
{
Vector<Option> list=new Vector<Option>();
Try
 {
String selectSQL="select id,heading from documents";
PreparedStatement pstmt=connection.prepareStatement(selectSQL);
ResultSet rs=pstmt.executeQuery();
while(rs.next()){
list.add(new Option(String.valueOf(rs.getLong(1)),rs.getString(2)));
}
} catch (Exception e)
 {
e.printStackTrace();}return list;
}
public void stateChanged(ChangeEvent e)
{
if(tabbedPane.getSelectedIndex()==2)
{
documentListVector=listDocuments();
documentList.setListData(documentListVector);
{
this.validate();}else if(tabbedPane.getSelectedIndex()==0){
documentListVector=listDocuments();
```

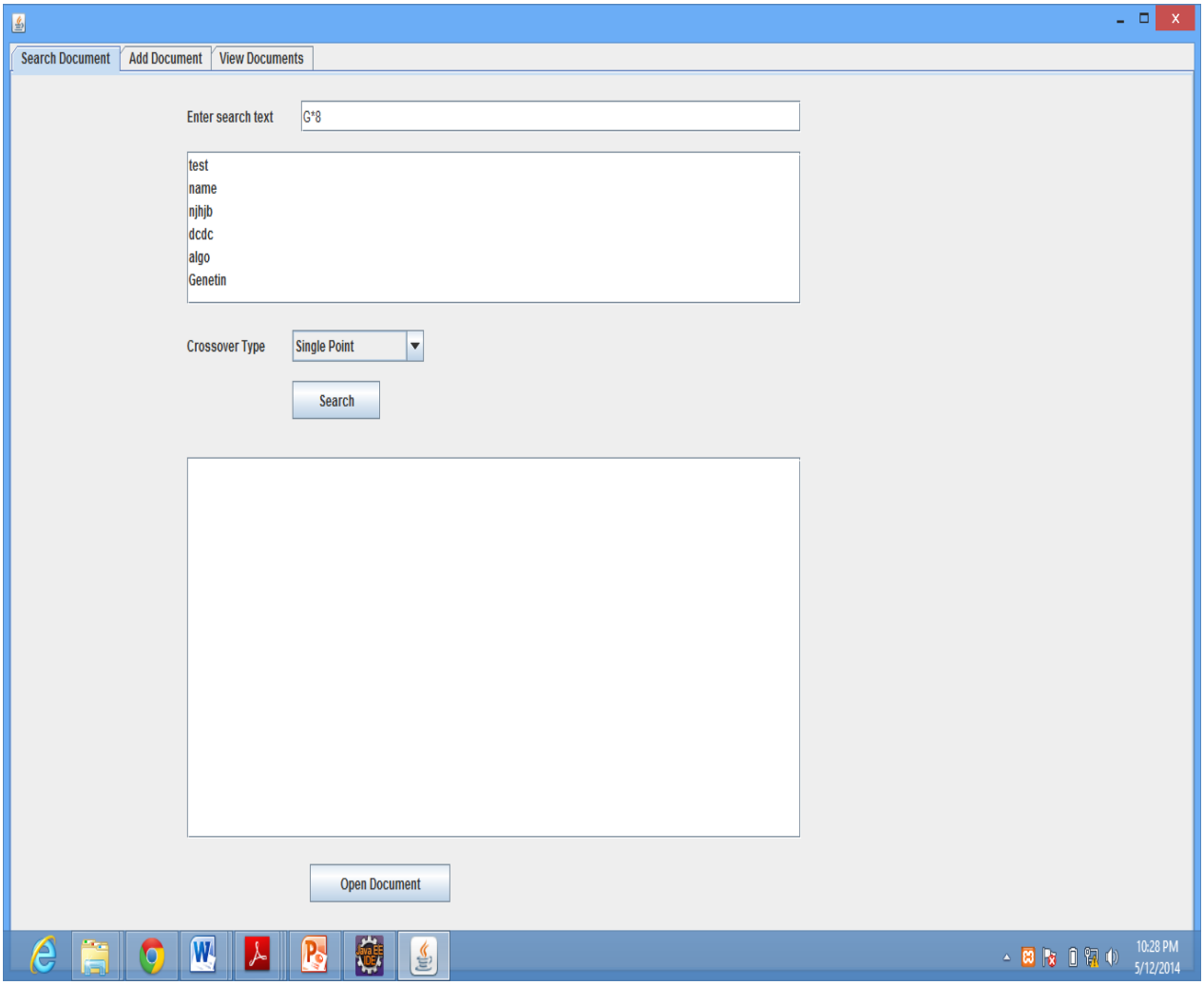
```
searchDocumentList.setListData(documentListVector);
this.validate();}}}
```

3.Option.java

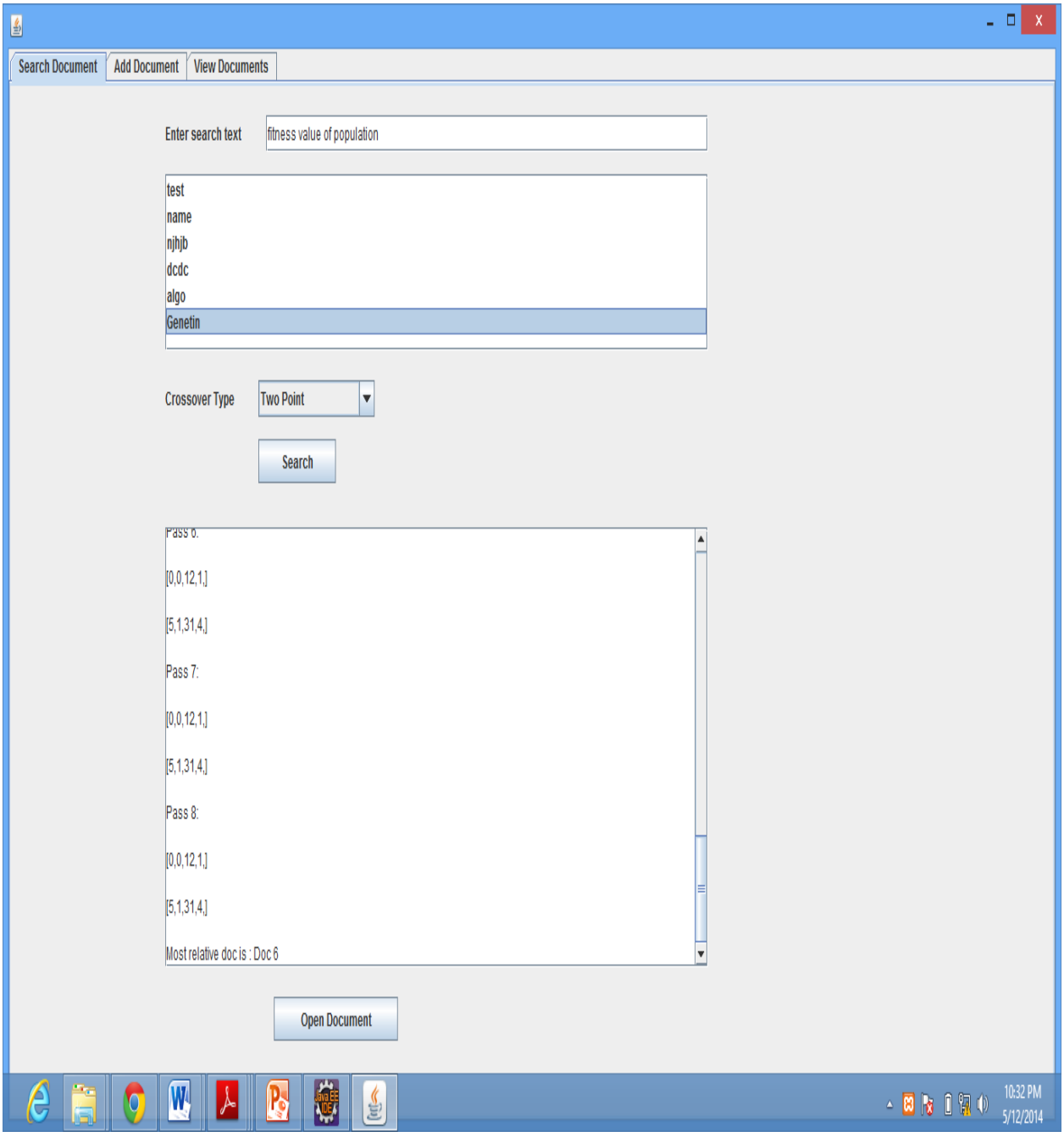
```
public class Option {
private String key=null;
private String value=null;
public Option(String key, String value) {
super();this.key = key;this.value = value;
        }
public String getKey() {
return key;}
public void setKey(String key) {
this.key = key;
}
public String getValue() {
return value;
}
public void setValue(String value) {
this.value = value;
}public String toString(){
return value;}
}
```

CHAPTER 7. SNAPSHOTS

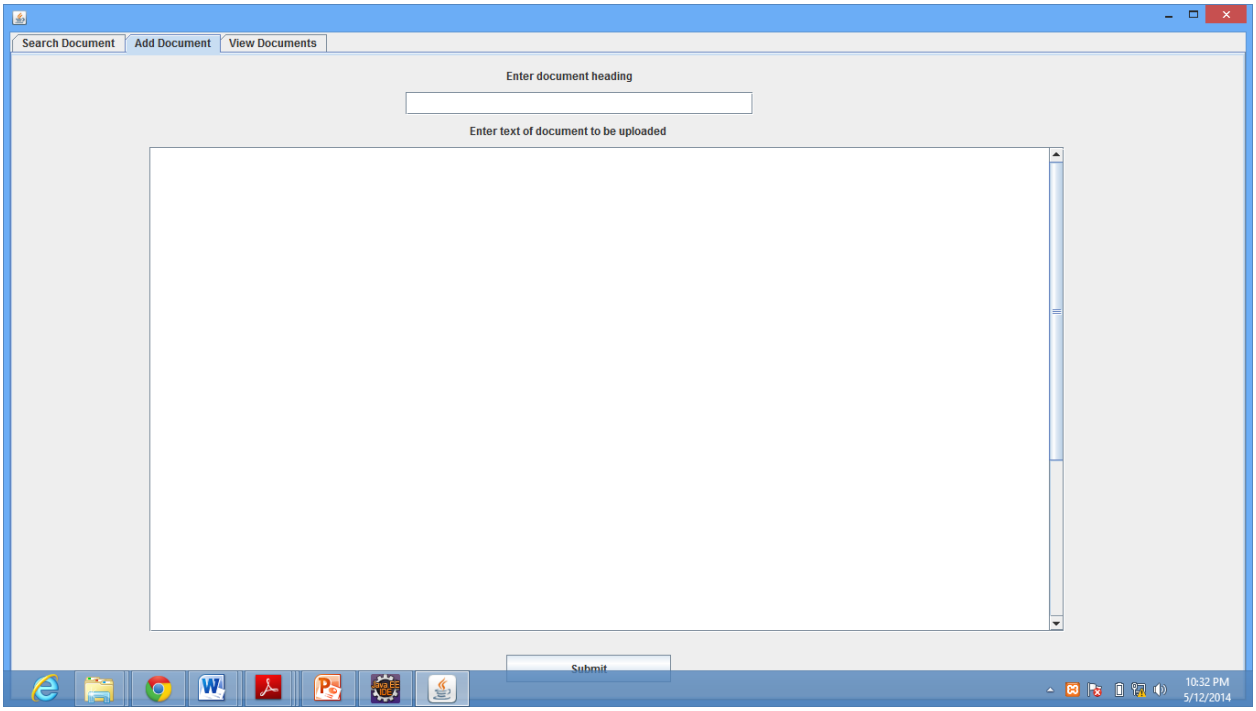
1.



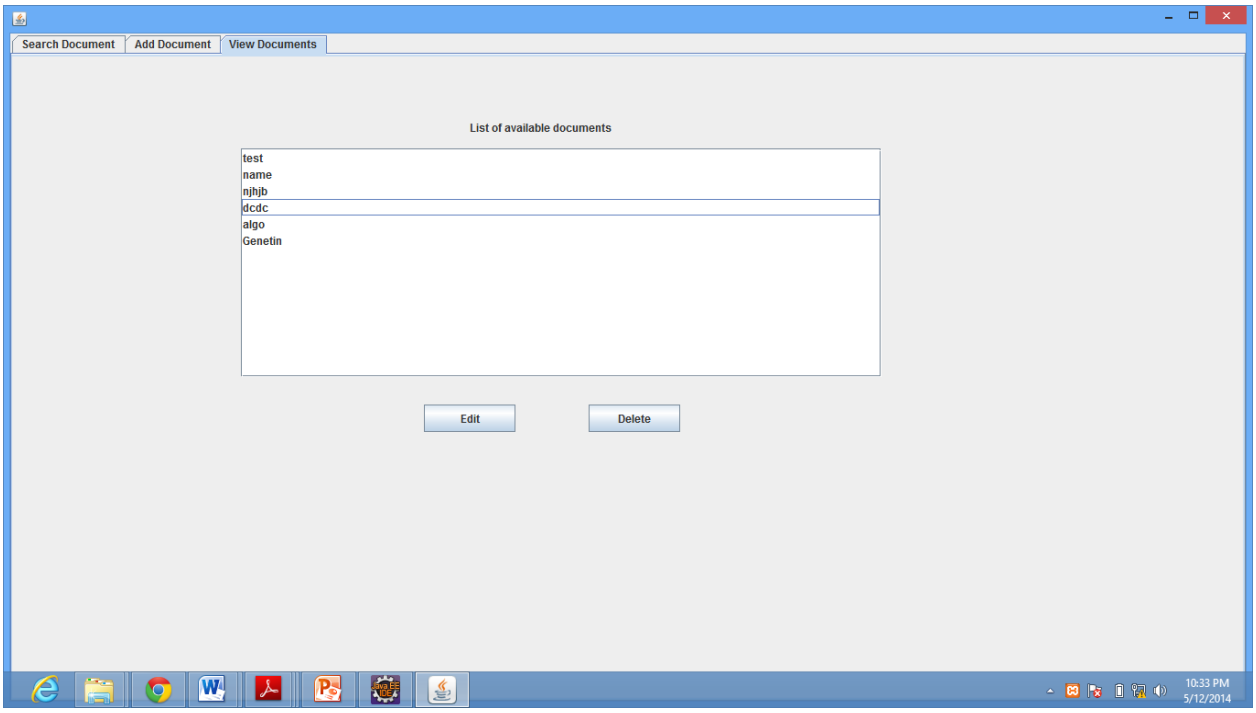
2.


Search Document
Add Document
View Documents
Enter search text
fitness value of population
test
name
njhjb
dcdc
algo
Genetin
Crossover Type
Two Point
Search
[0,0,12,1,]
[5,1,31,4,]
Pass 7:
[0,0,12,1,]
[5,1,31,4,]
Pass 8:
[0,0,12,1,]
[5,1,31,4,]
Most relative doc is : Doc 6
Open Document
10:32 PM
5/12/2014

3.


Search Document
Add Document
View Documents
Enter document heading
Enter text of document to be uploaded
Submit
10:32 PM
5/12/2014

4.


Search Document
Add Document
View Documents
List of available documents
test
name
njhjb
dcdc
algo
Genetin
Edit
Delete
10:33 PM
5/12/2014

REFERENCES

www.geatbx.com/docu/algindex-01.html#P153_5403

http://www.interscience.in/IJCNS_Vol1Iss4/39-44.pdf

http://www.cs.montana.edu/files/techreports/1112/Elser.pdf

http://cs.adelaide.edu.au/~zbyszek/Papers/p32.pdf

http://www.ra.cs.unituebingen.de/mitarb/streiche/publications/Introduction_to_Evolutionary_Algorithms.pdf

[Kaelbling *et al.*, 1996] L. Kaelbling, M. Littman, and A. Moore. Reinforcement learning: A survey. *Journal of Arti_cial Intelligence Research*, 4:237-285, 1996.

[Rennie and McCallum, 1999] Jason Rennie and Andrew McCallum. Using reinforcement learning to spider theWeb efficiently. In *ICML-99*, 1999.