APPLICATION OF GENETIC ALGORITHM IN MODELING UNIVERSITY ADMISSION DECISION SUPPORT SYSTEM

BY

AMINAT BOLATITO YUSUF

DEPARTMENT OF MATHEMATICS,

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

APRIL, 2014

APPLICATION OF GENETIC ALGORITHM IN MODELING UNIVERSITY ADMISSION DECISION SUPPORT SYSTEM

By

Aminat Bolatito YUSUF
M.Sc/Scien/06576/2009-2010

A THESIS SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTERS DEGREE IN COMPUTER SCIENCE.

DEPARTMENT OF MATHEMATICS,
FACULTY OF SCIENCE
AHMADU BELLO UNIVERSITY, ZARIA
NIGERIA

APRIL, 2014

DECLARATION

I declare that the work in this thesis entitled "Application of Genetic Algorithm in Modeling University Admission Decision Support System" has been performed by me in the Department of Mathematics. The information derived from the literature has been duly acknowledged in the text and list of references provided. No part of this thesis was previously presented for another degree or diploma at any other institution.

_____	_____	_____
Name of Student	Signature	Date

CERTIFICATION

This thesis report entitled APPLICATION OF GENETIC ALGORITHM IN MODELING UNIVERSITY ADMISSION DECISION SUPPORT SYSTEM by AMINAT BOLATITO YUSUF meets the regulation governing the award of the degree of M.Sc Computer Science of Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

_____ _____ Date _____
Chairman, Supervisory Committee

_____ _____ Date _____
Member, Supervisory Committee

_____ _____ Date _____
Member, Supervisory Committee

_____ _____ Date _____
Head of Department

_____ _____ Date _____
Dean, Postgraduate School

DEDICATION

In loving memory of my late mother, Mrs Yusuf Mulikat.

ACKNOWLEDGEMENT

My heart overflows with joy and gratitude to my Father and my God who strengthened me in the course of this research. I thank Him immensely for granting me journey mercies, provision and wisdom during this programme. From the depth of my heart, I sincerely appreciate my lecturers and supervisors, Dr. M. B. Hammawa (Major) and Dr. S. E. Abdullahi (Minor) who sacrificed their time to give me direction in the course of this research work. Thank you for your patience at the time when I displayed gross lack of understanding. May God who alone knows how best to reward every one, bless and reward you and your family.

My appreciation goes to the entire staff of the department of Mathematics, ABU Zaria, I say 'thank you' and especially to Dr. B. Sani (HOD, Mathematics Dept.) thank you sir for giving me access to the data used in this work and Dr. A.A. Obiniyi thanks sir for your support. Special thanks go to our Co-coordinator - Dr. A.M. Ibrahim and to Dr A. I.Yayaha

To my wonderful lecturer Mr Donfack Kana whose input cannot be quantified, thanks for all the support you gave me. To Mr Abdullahi Mohammed and Mr C. N. Inamani, thank you very much Sir.

My course mates, you are not left out - Sheba Alkali, Abubakar Hussaini, Sunday Adepoju, Abdulrazaq Abdulraheem, Hajiha Bazama and Aliyu Salisu thank you all for your encouragement.

To my wonderful family, daddy "Mr Bankole Yusuf" I love you daddy and I miss home. Thanks for your understanding. To my beloved siblings Zikurullahi Yusuf, Ibrahim Yusuf and Abosede Basir thanks for allowing me to further my study.

My colleague Amen Agbowange thanks for being there for me each time I need to take a break.

I sincerely thank the management of Ahmadu Bello University, Zaria for giving me this opportunity to further my studies.

ABSTRACT

This thesis evaluates the impact of Universities' entry qualification requirements and the performance of students, in order to decide how these factors bring patterns that are best for admission. The proposed model uses the defined University's entry qualification as input variables and the students' first year performance to determine the best admission requirements. Genetic algorithm was used as the searching technique to determine the hidden relationship between the input and the associated performance. Students' admission data and their corresponding first year results were obtained from the department of Mathematics, Ahmadu Bello University, Zaria. The results indicated that the observed performance of students whose admission into Mathematics Department through the University Matriculation Examinations, Post University Tertiary Matriculation Examinations and O'levels depends more on their respective mathematics and physics average performance in all the three examinations than their entry scores in the individual examination. A comparative study using a statistical model show that the result obtained from the genetic algorithm approach were in line with the result of the statistical model. The model was implemented using java programming language, developed in Netbean environment.

TABLE OF CONTENT

DECLARATION iv

CERTIFICATION v

DEDICATION vi

ACKNOWLEDGEMENT vii

ABSTRACT ix

TABLE OF CONTENT x

CHAPTER ONE 1

GENERAL INTRODUCTION 1

1.1 Introduction 1

1.2 Background Information 3

1.3. Problem Definition and Motivation 5

1.4. Objective of the Study 6

1.5 Research Methodology 6

1.6 Contribution to Knowledge 7

1.8 Significant of the Study 9

1.9 Organization of the Thesis 9

CHAPTER TWO 10

REVIEW OF LITERATURE 10

2.1 Introduction 10

2.2 History of Evolutionary Algorithms 10

2.2.1. Search Techniques 12

2.2.2 Evolution Theory and Genetic Algorithm 15

2.3 Genetic Algorithms 16

2.3.1 Applicability of Genetic Algorithm 20

2.4 University Education in Nigeria 21

2.5 Related Work 23

2.6 Literature Gap 31

CHAPTER THREE 33

MODELING THE STUDENT'S PERFORMANCE DECISION SUPPORT SYSTEM 33

3.1. System Description 33

3.2. Benchmarking Student's Performance 34

3.3 Features Extraction and Normalization of Data 35

3.3.1 Handling Discrepancy of Feature Extraction 35

3.3.2 Handling of Normalization 39

3.4 Genetic Algorithm 42

3.4.1. Fitness Measurement 45

3.4.2 Selection 46

3.4.3 Crossover 46

3.4.4 Mutation 46

3.5 Basic Genetic Algorithm Procedure 47

3.5.1 Initial Population Generation 47

3.5.2 Fitness Evaluation 49

3.5.3. New Population 49

3.5.4 Acceptance 53

3.5.5 Termination Condition 54

CHAPTER FOUR 55

IMPLEMENTATION OF THE DECISION SUPPORT SYSTEM 55

4.1 Introduction 55

4.2 System Requirement 55

4.2.1 Hardware Requirement 55

4.3.1 The Data Use for Evaluation 55

4.3.2 Data Normalization 56

4.3.3 Data Storage 56
4.4 System Implementation 56
4.4.1 Upload Unit 57
4.4.2 Subjects Selection Unit 59
4.4.3 Selection Unit 60
4.4.4 Searching Unit 61
4.5 Result and Discussion 63
4.6. Validation 66
4.7 Conclusion 68
CHAPTER FIVE 69
SUMMARY, CONCLUSION AND RECOMMENDATION 69
5.1 Summary 69
5.2 Conclusion 69
5.3 Recommendations 70
REFERENCES 71
APPENDIX 80

List of Figures

Figure: 2.1 Classes of Search Techniques (Goldberg, 1989) 13

Figure 3.1: System modules relationship 33

Figure 3.2: Representation of student's subject in each examinations.....34

Figure 3.3: Genetic Algorithm flowchart 44

Figure 4.1: The system main interface 57

Figure 4.2: Faculty, department and the session of year of data selection.....58

Figure 4.3: Report of successful upload of file.....58

Figure 4.4: Interface prompting user for more upload 59

Figure 4.5: System interface for subject selection 60

Figure 4.6: System interface for population size 61

Figure 4.7: System interface for searching.....62

Figure 4.8 : Output result of best pattern.....64

Figure 4.9: Graphical display of average score for each subject 65

Figure 4.10: Graphical result of statistical analysis 68

List of Tables

Table 3.1: Grade conversion table…….40

Table 3.2: Grouping of CGPA into different classes…….45

Table 3.3: An illustrating example of students' classifier rules with their CGPA. ...47

Table 3.4: CGPA grouping and the unit points…….. ...49

Table 3.5: Fitness table…….. 49

Table 3.6: Illustration of the proportional selection method…….50

Table 3.7: The best combinations selected for mating…….. 50

Table 3.8: pairing classifiers for crossover…….51

Table 3.9: crossover to form new rules……. 52

Table 3.10: mutated classifier……... 53

Table 4.1: Table of average score performance…….65

Table 4.2: PUTME, UTME and SSCE correlation relationship……....66

Table 4.3: Result from statistical value using descriptive Statistics……. 67

CHAPTER ONE

INTRODUCTION

1.1 Introduction

University education in Nigeria has witnessed tremendous development since the country's independence in 1960 (Adeyemi, 2010). This is in recognition of the fact that the national policy of education stipulates that university education in Nigeria shall make optimum contribution to national development by intensifying and diversifying its programmes for the development of high level manpower within the context of the needs of the nation (FGN, 2004). One of the factors limiting the University in performing its roles as it is required is the quality of students admitted into various academic programmes (Adeyemo and Kuye, 2006). It is expected that an average student admitted into the University should be able to face academic studies with ease and pass his/her courses without engaging examination malpractice; because it is assumed that such student would have had prior experience in public examination.

Students are expected to have sat for the Senior Secondary School Certificate Examinations (SSCE) and passed the minimum requirement and presented themselves for the Joint Admission and Matriculation Board (JAMB) Examination as a selection test and pass at acceptable cut-off point before being offered admission into the university (Salim, 2006). Despite these public examinations that the Nigerian undergraduates had gone through, it has been observed that their performances in the first two years of their undergraduate studies do not usually match that of the JAMB which is used as the basis for their admission in the first place into the University (Adeyemo and Kuye, 2006). Many students hardly pass all their first year courses and

majority of those who successfully do so usually have poor grades. A great percentage of university graduates in Nigeria fall below second class upper division and the number of spillover students in various departments are equally high. The situation gets worse as those who manage to graduate are not productive in the labour market because they are unable to meet the expectations of the employers (Ajala, 2010).

In 2005, Universities recommended that further screening be conducted on candidates who sat scored between 180 and 200 marks in the university matriculations examination depending on the university where admission is being sought. The recommendation was made with the hope that the post- JAMB screening exercise would restore the past glory of tertiary education in the country and make university education accessible only to those who want and need it (Ande, 2006). Hence, these lead to multiple admissions selection criteria.

The issue of whether or not the scores in O'levels, UME and PUTME correlate to the candidate's performance in the university, especially in the first year has begun to attract researchers' attention. Some researchers claimed that O'levels and UME have no correlation. This needs further investigation and evaluation in order to arrive at a reasonable conclusion. It is known that the selection of students is a complex decision making process, in which multiple selection criteria often needs to be considered. However, the selection criteria used in higher education admission processes varies widely among programmes and no consistent conclusions can be reached on the predictive values of these criteria (Wilson, 1999). Statistical procedures, such as discriminant analysis and regression analysis are traditionally used for predicting the potential academic success of the applicant (Graham, 1991). In the world of information

processing, there are lots of data with increasingly complex multi-domain problems containing either real-world or computer-generated data, which the statistical data processing tools may not be sufficient enough to handle, hence a more advanced approach needs to be developed.

This thesis uses genetic algorithm to evaluate the admission requirement into various university programmes using computer science of the department of mathematics, Ahmadu Bello University Zaria as a case study. The system predicts the patterns that are suitable for selection of students' into the programme. Genetic algorithm has shown a promising feature in the area of decision support system. The principle of survival of the fittest in which genetic algorithm was modeled, could be of great benefit in the process of random selection from the available data.

1.2 Background Information

Despite stringent measures and strategies employed by the Nigerian government to ensure that educational standards are maintained at least at university level, students who after passing through these vigorous examinations still perform far below expectations. For instance, from the summary of Computer Science students' data, Mathematics department, Ahmadu Bello University, Zaira for 2009/2010 session, out of 173 students that were admitted into the programme none had CGPA above 4.5, 18 had CGPA between 3.5 and 4.49, 35 had CGPA between 2.4 and 3.49, 10 had CGPA between 1.5 and 2.39. At the end 97 student were recorded to have one or two carry over and 2 were asked to withdraw (Departmental second semester summary, 2010). This implies that only 10.78% of the students actually had satisfactory results at the end of their stay of first academic year. This also shows that 87.22% of the students had academic challenges as undergraduate students. The high rate of poor academic

achievement among undergraduate is not unconnected with the channel through which they gained entry into the University. Ebiri (2010), observed that using JAMB as a yardstick for admission of students into Nigerian universities has led to the intake of poor caliber of candidates, characterized by high failure rate, increase in examination malpractice, high spillovers and the production of poor quality output that are neither self-reliant nor able to contribute effectively in the employment world.

Ironically, the process of selecting candidates for admission into tertiary institutions has largely depended on some fixed combinations of some subjects taken by applicants in their lower level classes. However, this technique has never been proved efficient in admitting candidates that may perform well in the chosen courses. The fast growing of candidates seeking for admission into tertiary institutions, there is a need to use past data for decision support in admitting suitable candidate for a course of study. Universities are facing the immense and quick growth of the volume of educational data (Schönbrunn and Hilbert, 2006). Intuitively, this large amount of raw stored data contains valuable hidden knowledge, which could be used to improve the decision-making process of universities (keshavamurthy *et al.*, 2010). An analysis of the existing transaction data provides the information on students that will allow the definition of the key processes that have to be adapted in order to enhance the efficiency of studying (Mario *et al.*, 2010). It is tedious and difficult to analyze such large voluminous data and establishing relationship between multiple features manually. Our proposed system delves into the problem of finding data patterns in admission datasets and provides a technique to predict the performance of students in the first year in the University based on the admission combination.

1.3. Problem Definition and Motivation

Higher education systems all over the world nowadays are challenged by the new information and communication technologies (Boufardea and Garofalakis, 2012). Moreover, with the increase in competition among the prospective students into higher institutions, most Universities are facing the daunting task of selecting the best students, who have the ability and skills to pursue and succeed in their academic career in a particular field of studies. This is because Universities are interested in increasing performance. Performance is one of the means of measuring University's quality and reputation (Jusoff *et al.*, 2008), thus higher institutions are becoming more interested in predicting the paths of students, and identifying which students will require assistance in order to graduate (Luan, 2004). In order to be able to achieve this objective, the finding relationships and patterns that exist but are hidden among the vast amount of educational data is needed. This knowledge will help in educational main processes such as counseling, planning, registration, and evaluation in order to give suitable recommendation of the students.

Predictions of qualities of entry result that should be used in admitting students into respective programmes are published in Nigeria, mostly in medicine, education and engineering and most of these are done using statistical approaches. The work of Adewale *et al* (2007) and Luna (2004) show a great insight that the field of computer science has a lot to offer in contributing to the knowledge evaluation and the effectiveness of JAMB-UME Scores, post-UME scores and SSCE Scores.

The aim of this thesis is to determine how aggregation of UME, post-UME and SSCE scores bring a pattern that is commonly attributed to the good performance of first year students' academic achievement at the university in the department of Mathematics, Ahmadu Bello University, using the concept of Genetic Algorithm. The identification of

these patterns can help in the selection process for admitting students into the various departments.

1.4. Objective of the Study

The main objective of this thesis is to design a model using genetic algorithm that can be employed in searching trends or pattern in student's previous admission records. This is achieved by using the aggregation of UME, Post UME, O'level scores against their corresponding CGPA at the end of their first academic year in the University. Realizing this objective can help in candidates' selection criteria for admission process into the university. This main goal can be achieved by means of the following objectives which are:

1. To determine the means by which data collected can be translated to meaningful ones.
2. To develop a model for searching hidden pattern among the available data set using genetic algorithm.
3. To implement model of genetic algorithm
4. To test and validate the model using real data of students' records.

1.5 Research Methodology

In designing the system, the objectives stated in section 1.4 can be achieved by considering the following steps:

1. Data collected are subjected to the process of feature extraction and normalization. The data gathering process involves the collection of raw data about students, which include the UTME score, PUTME score and O'level results (which are the entry requirements into the University). Feature extraction is carried out since the data collected can be inconsistence, incomplete or noisy.

This may be as a result of a number of factors ranging from data entry or transmission problem, discrepancy in the naming convention, duplicated records or incomplete data or removal of unwanted entries that are not required. All these affect the analysis. The data used for the analysis was entered into an Excel spreadsheet file. Each student was being identified using his/her JAMB number. Also, normalization of data is carried out on each subject's grade by using a uniform data representation since each examination is being graded differently. These datasets are stored and accessed using Mysql relational database.

2. The model of genetic algorithm of principle of survivor of the fittest is used in searching through the formatted student academic performance. The different operators of GA perform their own work by following the instruction for the GA, till the best patterns are found.
3. The model is implemented using java with Mysql relational database which provide the required functionalities in holding students' academic data.
4. The performance of the model is then compared with the statistical approach using SPSS software.

1.6 Contribution to Knowledge

Using decision support system for admission selection process by genetic algorithm, the University admission requirement is validated against the performance at the end of their first academic year, to detect hidden trends among the students' performance. The contributions of the study to knowledge are outlined below:

1. Government's policy to promote higher education, learning and research will be realized since the right candidates are selected and trained in the universities this

will bring about the production of the right human resources who are the major factors of production.

2. An efficient, detailed and unbiased procedure of using average performance for admission into universities is put in place as against using single subject performance for selection processes.
3. Selection of best students for university education will also make teaching and learning easier as the best student is usually an individual who is focused and disciplined. This will go a long way in making the goal of education achieved effectively for economic growth and development in to the various sectors of the nation.
4. Since it provides better admission opportunity for qualified candidates, better qualified graduates will now be turned out into the job market as opposed to the output that comes from persons who struggle through the universities because they were never qualified to be there in the first place.

1.7 Scope of the Study

This research work is a case study of Computer Science, Mathematics Department of Ahmadu Bello University, Zaria, Kaduna State. The research aim to cover records of students admitted in three years academic session into 100 level through O'level, JAMB and Post-JAMB scores respectively. This study therefore grew out of curiosity to find out how prediction helps to identify and to improve students' performance.

To the best of my knowledge, no study in the literature at my disposal has been carried out to compare the academic achievement between undergraduate students admitted through their O'level, Post-JAMB and JAMB scores in Ahmadu Bello University, Zaria. The statement of the problem therefore seeks to identify best patterns at which the

aggregation of the three examinations brings in the first year performance of the student. This early prediction allows the instructor to provide appropriate advising or select those with less risk for admission.

1.8 Significant of the Study

Precisely, the significance of this study is based on

1. Determining the extent to which scores in examinations conducted by the West African Examination Council (WASSCE), National Examinations Council (SSCE) and in conjunction with the Joint Admissions and Matriculation Board (UME) and post-UTME to predict future academic achievement of students in university degree examinations.
2. Develop structural models for predicting the academic achievement in university degree examinations based on performance in public examinations.

1.9 Organization of the Thesis

The thesis is organized as follow: in the second chapter, review was made on the predictive technique using Genetic Algorithm and literatures that were accomplished in the area of University admission variables were reviewed. In the third chapter, description was made on how to model finding patterns in admission combinations, by first normalizing and later applying Genetic Algorithm. In the fourth chapter, a proposed implementation of the decision support system is designed as followed from chapter three, using java programming language and Msql relational database. The thesis concludes in chapter five, with summary, conclusion and recommendation.

CHAPTER TWO

REVIEW OF LITERATURE

2.1 Introduction

In this chapter, a review is made of predictive technique in genetic algorithm. A description of the research that has been accomplished in the area of university admission decision making is also provided.

2.2 History of Evolutionary Algorithms

In the 1950's and 1960's several computer scientists independently studied evolutionary systems with the idea that evolution could be used as an optimization tool for engineering problems. The idea in all these systems was to evolve a population of candidate solutions to a given problem, using operators inspired by natural genetic variation and natural selection.

In the 1960s, Rechenberg introduced "evolution strategies" as a method he used to optimize real valued parameters for devices such as airfoils. This idea was further developed by (Schwefel, 1977). The field of evolution strategies has remained an active area of research, mostly developing independently from the field of genetic algorithm (although recently the two communities have begun to interact). Fogel *et al* (1966) developed "evolutionary programming," a technique in which candidate solutions to given tasks were represented as finite state machines, which were evolved by randomly mutating their state transition diagrams and selecting the fittest. Together, evolution strategies, evolutionary programming, and genetic algorithms form the backbone of the field of evolutionary computation.

Several researchers working in the 1950s and the 1960s developed evolution inspired algorithms for optimization and machine learning. Box (1957), Friedman (1959), Bledsoe (1961), Bremermann (1962), and Reed *et al* (1967) all worked in this area, though their work has been given little or none of the kind of attention or follow up that evolution strategies, evolutionary programming, and genetic algorithm have seen. In addition, a number of evolutionary biologists used computers to simulate evolution for the purpose of controlled experiments (Baricelli 1957, 1962; Fraser 1957; Martin and Cockerham 1960). Evolutionary computation was definitely in the air in the formative days of the electronic computer.

Genetic algorithm (GA) was invented by John Holland and his students and colleagues at the University of Michigan in the 1960s and the 1970s (Holland, 1975). In contrast with evolution strategies and evolutionary programming, Holland's original goal was not to design algorithm to solve specific problems, but rather to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems.

Holland's 1975 book “Adaptation in Natural and Artificial Systems” presented the genetic algorithm as an abstraction of biological evolution and gave a theoretical framework for adaptation under the GA. Holland's GA is a method for moving from one population of "chromosomes" (for example, strings of ones and zeros, or "bits") to a new population by using a kind of "natural selection" together with the genetics inspired operators of crossover, mutation, and inversion. Each chromosome consists of "genes" (for example, bits), each gene being an instance of a particular "allele" (for example, 0 or 1). The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and on average the fitter chromosomes produce more offspring than the less fit ones. Crossover exchanges subparts of two chromosomes, roughly

mimicking biological recombination between two single chromosome ("haploid") organisms; mutation randomly changes the allele values of some locations in the chromosome; and inversion reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed.

Holland's introduction of a population based algorithm with crossover, inversion, and mutation was a major innovation. Unlike Rechenberg's evolution strategies which started with a "population" of two individuals, one parent and one offspring, the offspring being a mutated version of the parent; many individual populations and crossover were not incorporated until later. (Fogel *et al.*, 1966) evolutionary programming likewise used only mutation to provide variation. Moreover, Holland was the first to attempt to put computational evolution on a firm theoretical footing (Holland, 1975). Until recently this theoretical foundation, based on the notion of "schemas," was the basis of almost all subsequent theoretical work on genetic algorithms.

In the last several years, there has been widespread interaction among researchers studying various evolutionary computation methods, and the boundaries between GAs, evolution strategies, evolutionary programming, and other evolutionary approaches have broken down to some extent. Today, researchers often use the term "genetic algorithm" to describe something very far from Holland's original conception (Melanie, 1999).

2.2.1. Search Techniques

In the world of information processing, we are faced with increasingly complex multi-domain problems containing either real-world or computer-generated data. To consider

these problems the classical data processing tools may not be sufficient and more advanced approaches need to be developed (Wafa, 2013). A unified approach based on searching techniques may be adopted for easy search of optimum solution. Search techniques in general, as illustrated in Figure 2.1 can be grouped into three broad classes: Calculus based Enumerative and Guided Random search (Goldberg, 1989).

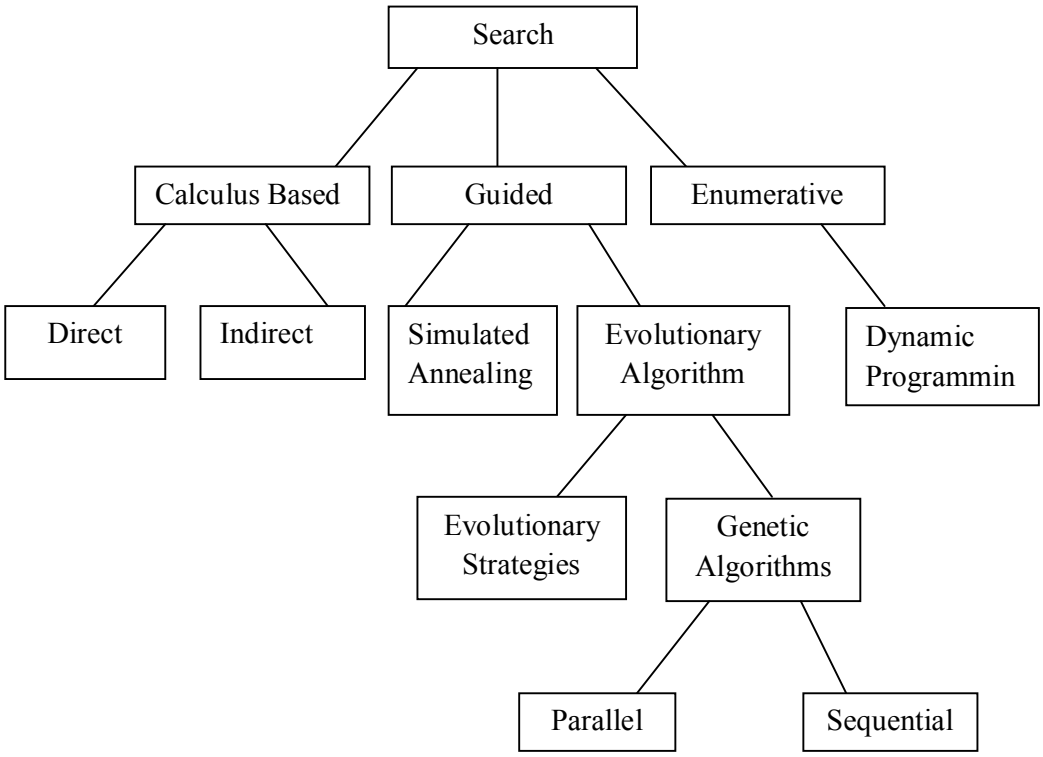


Figure: 2.1 Classes of Search Techniques (Source: Goldberg, 1989)

1. Calculus based techniques: use a set of necessary or sufficient conditions to be satisfied by the optimal solutions of an optimization problem. These techniques are sub-divided into indirect and direct methods. Indirect methods look for local extrema by solving the usually non-linear set of equations resulting from setting the gradient of the objective function equal to zero. The search for possible solutions (function peaks) starts by restricting the search to points with zero slope in all directions. Direct methods, such as Newton and Fibonacci, seek

extrema by "hopping" around the search space and assessing the gradient of the new point, which guides the direction of the search. This is simply the notion of Hill-Climbing which finds the best local point by "climbing" the steepest permissible gradient. However, these techniques can only be employed on a restricted set of "well behaved" problems (Bandyopadhyay and Saha, 2013).

2. Enumerative techniques search: every point related to an objective function's domain space (finite or discretized), one point at a time. They are very simple to implement but may require significant computation. The domain space of many applications is too large to search using these techniques. Dynamic programming is a good example of an enumerative technique (Bandyopadhyay and Saha, 2013).
3. Guided Random search techniques: are based on enumerative techniques but use additional information to guide the search. They are quite general on their scope, being able to solve very complex problems. Two major sub-classes are: Simulated Annealing and Evolutionary Algorithms, although both are evolutionary processes. Simulated Annealing uses a thermodynamic evolution process to search minimum energy states. Evolutionary Algorithms, on the other hand, are based on natural selection principles. This form of search evolves throughout generations, improving the features of potential solutions by means of biological inspired operations. These techniques sub-divide, in turn, into evolution strategies and genetic algorithms. Evolution strategies were proposed by Rechenberg and Schwefel in the early seventies. They present the ability to adapt the process of "artificial evolution" to the requirements of the local response surface. This means that evolution strategies are able to adapt their major strategy parameters according to the local topology of the objective

function. This represents a significant difference to traditional GA. Following Holland's original Genetic Algorithm proposal, many variations of the basic algorithm have been introduced. However, an important and distinctive feature of all GA is the population handling technique. The original GA adopted a generational replacement policy where the whole population is replaced in each generation. Conversely, the steady-state policy used by many subsequent GA employ a selective replacement for the population. It is possible, for example, to keep one or more individuals within the population for several generations, while those individuals sustain a better fitness than the rest of the population (Bandyopadhyay and Saha, 2013).

2.2.2 Evolution Theory and Genetic Algorithm

According to biology, all living organisms consist of cells and each cell contains one or more chromosomes. A chromosome can be conceptually divided into genes, each of which encodes a particular protein. Organisms' chromosomes can be arrayed in pairs called diploid or unpaired called haploid. In nature, most sexually reproducing species are diploid, including human beings, whose each have 23 pairs of chromosomes in the cell of the body. During sexual reproduction, recombination (or crossover) occurs in each parent, genes are exchanged between each pair of chromosomes to form a gamete (a single chromosome), and then gametes from the two parents pair up to create a full set of diploid chromosomes. In haploid sexual reproduction, genes are exchanged between the two parents and offspring are subject to mutation, in which single nucleotides (elementary bits of DNA) are changed from parent to offspring. The fitness of an organism is typically defined as the probability that the organism will live to reproduce (viability) or as a function of the number of offspring the organism has (fertility).

In genetic algorithms, the term chromosome typically refers to a candidate solution to a problem, often encoded as a bit string. The "genes" are either single bits or short blocks of adjacent bits that encode a particular element of the candidate solution. Crossover typically consists of exchanging genetic material between two single chromosome haploid parents. Mutation consists of flipping the bit at a randomly chosen locus (or, for larger alphabets, replacing the symbol at a randomly chosen locus with a randomly chosen new symbol). Most applications of genetic algorithms employ haploid individuals, particularly, single chromosome individuals (Melanie, 1999).

2.3 Genetic Algorithms

Genetic algorithm (GA) refers to a model introduced and investigated by John Holland in 1975 for adaptation processes of nature. Generally stated, a GA is any population based model that uses selection and recombination operators to generate new sample points in a search space. GA is a global search procedure that searches from one population of points to another and a probabilistic search procedure, which is being frequently applied to difficult optimization and learning problems. There are two versions of the GA, namely the natural GA and the computational GA.

Genetic algorithms were inspired by the processes observed in natural evolution. They attempt to mimic these processes and utilize them for solving a wide range of optimization problems. In general, genetic algorithms perform directed random searches through a given set of alternatives with respect to the given criteria of goodness. These criteria are required to be expressed in terms of an objective function, which is usually referred to as a fitness function.

Genetic algorithms require that the set of alternatives to be searched through be finite. If we want to apply them to an optimization problem where this requirement is not satisfied, the set involved and select an appropriate finite subset. It is further required

that the alternatives be coded in strings of some specific finite length which consist of symbols from some finite alphabet. These strings are called chromosomes, the symbols that form them are called genes, and their set is called a gene pool. Genetic algorithms search for the best alternative in the sense of a given fitness function through chromosomes evolution (Sapna *et al.*, 2012). The basic cycle's steps in genetic algorithms are:

1. Creation of a "population" of strings,
2. Evaluation of each string,
3. Selection of "best" strings, and
4. Genetic manipulation, to create the new population of strings.

Creation of a population of strings

The basis of genetics in nature is a chromosome (string). When applying this thought to computer science and genetic algorithms, each individual in the search space, i.e. each solution to the problem at hand, needs to be encoded so that it can be thought of as a chromosome. Each element of the population is encoded into binary coded or real (decimal) coded.

Evaluation of each string

The performance (or fitness) of each individual of the population is evaluated, with respect to the constraints imposed by the problem. . Fitness evaluates all the candidate solutions and the quality of all individual solutions and gives a criterion to rank candidate solutions which is the basis of making a decision as to whether a particular individual solution is fit to survive and reproduce. The fitness measure steers the evolutionary process towards better approximate solutions to the problem. It takes in one chromosome at a time as input and returns a single numeric value, which is

indicative of the ability or utility of the candidate solution represented by the input chromosome. Fitness of individuals in a population can be measured in many ways. It can be measure of error between the original and desired output of a solution. It can be compliance of the structure to the task it is required to solve based on a user specified criteria.

Selection

At every iterations, chromosomes are recombined to create new chromosomes in an attempt to find better chromosomes. As genetic algorithms follow the theory of natural evolution, better individuals should be able to survive and reproduce. The selection operator will determine the individuals that will survive to the next generation, and should thus be defined so that the ones with the best fitness are more likely to survive in order to increase the average fitness of the population.

The simplest way of defining a selection operator is to use a purely elitist selection. This selects only the "elites", for example., the individuals with the highest fitness. Elitist selection is easy to understand and simple to implement; one can simply discard the weakest individuals in the population. However, elitist selection isn't the best choice, as it may very well result in getting stuck to a local optimum.

Another and a more common way of defining the selection operator is to use a fitness proportionate selection, which can be implemented with a "roulette-wheel" sampling (Melanie, 1999). Here, each individual is given a slice of the "wheel" that is in proportion to the "area" that its fitness has in the overall fitness of the population. This way, the individuals with higher fitnesses have a larger area in the wheel, and thus have a higher probability of getting selected. The wheel is then spun for as many times as there are individuals needed for the population. In general, a fitness-proportionate selection operator can be defined by assigning a probability of surviving, to each

individual, with coefficient to ensure that individuals with better fitness values are more likely to be selected. Comparing the actual values given by the fitness function is difficult, so these actual values should be used as coefficients with caution. However, by examining the order of fitnesses it is possible to employ the idea of survival of the fittest by having a linear relation between the order of fitness and the coefficient. A common selection operator is a crossing of the two methods presented above; the survival of the very fittest is guaranteed by choosing the best individual with elitist methods, while the rest of the population is selected with the probabilistic method in order to ensure variety within the population.

Some researchers also use the tournament technique to select the next generation (Seng *et al*., 2005). In this type of selection, a tournament is conducted among few individuals chosen randomly from the population. The winner or best member is selected as a result of a tournament. The tournament size determines how many random members are selected for the tournament. Tournament size determines the selective pressure; large tournament size favors fitter solutions for selection.

Crossover

One of the most important operators in GA is crossover. Crossover is a means for two strings to produce two offspring by mixing and matching their desirable qualities through a random process. After reproduction, crossover proceeds into two steps:

1. Two strings are selected
2. Segments of each string are chosen at random (segment length and location) and the information contained in these segments is exchanged between the two strings.

Several methods can be used for choosing the length and location of exchange sites. These could be one point, two point and uniform crossover point (Antonio, 2013).

Single-point crossover: After reproduction, crossover may proceed in two steps. First, members of the newly reproduced chromosomes in the mating pool are mated at random. Second, each pair of chromosomes undergoes crossover as follows: an integer position k along the chromosome is selected uniformly at random between 1 and the chromosome length l. Two new chromosomes are created swapping all the genes between k+1 and l.

Two-point crossover: Two-point cross over is very similar to single-point crossover except that two cut-points are randomly generated instead of one.

Uniform crossover: In uniform crossover, a value of the first parent's gene is assigned to the first offspring and the value of the second parent's gene is to the second offspring with a probability value P_c.

With probability P_c the value of the first parent's gene is assigned to the second offspring and the value of the second parent's gene is assigned to the first offspring,

Mutation

Mutation is intended to prevent early convergence of all solutions in the population into a local optimum of the solved problem. The mutation operation randomly changes the offspring resulted from crossover. The mutation operator produces small random changes to the bit string by choosing a single bit at random, then changing its value (Takale, 2004).

2.3.1 Applicability of Genetic Algorithm

Genetic Algorithms are applied to a very broad range of problems in a variety of industries including financial modeling, manufacturing, data mining, engineering, design and science. Some examples are:

1. Traveling salesmen problems such as vehicles routing
2. Scheduling problems such as multiprocessor scheduling (Kaur *et al.*, 2010)

3. Fault prediction modeling for web application (Marchima *et al.*, 2011)
4. Plucking problems such as shipping container operation
5. Finger prints registration problem (El-Emary *et al.*, 2008)

Genetic Algorithms are well suited to classes of difficult problems where an analytical or algebraic solution has not been identified or is computationally infeasible.

2.4 University Education in Nigeria

The history of University education in Nigeria can be traced back to the setting up of the Elliot Commission by the then British Colonial Government in 1943. The commission recommended among others, the establishment of a university college in Nigeria. This resulted in the founding of University College, Ibadan and University of Nigeria, Nsukka in 1948 and 1960 respectively. The implementation of Lord Eric Ashby's commissions' report in 1962 started the country's march towards general access to University Education (Salim, 2006). In 1962, Ahmadu Bello University, Zaria, University of Ife (now Obafemi Awolowo University), Ile Ife and University of Lagos were established. University of Benin was established in Benin City in 1970; initially as a College of Technology. It is important to note at this point that of the first six universities established in Nigeria before government took over all schools in 1974, only two, universities of Ibadan and Lagos were Federal Institutions. The remaining four: Ahmadu Bello, Ife and Nsukka were regional, state and later Federal Institutions at one time or the other.

The further increase in the demand for University education led to the establishment of seven new universities in 1975 by the then Military Administration, these seven new universities were sited at Calabar, Ilorin, Jos, Kano, Maiduguri, Port Harcourt and Sokoto. From the period of return to democratic rule in 1979 through the military intervention in 1983 and to another return to democratic rule in 1999, the Federal, State,

even individuals and corporate bodies have established and are still establishing new universities. After the establishment of the earlier universities, no thought was given to the possibilities of co-operative action in the matter of admission of students into universities. As a result of this position, every one of the existing universities conducted its own entrance examination and admitted its own students. Salim (2006) noted that after some time, these systems of admission revealed serious limitations and waste of resources in the process of administering the entrance examinations.

One of the major limitations of the individual university admission procedure was the issue of multiple applications and multiple admissions inherent in the system then. It was possible in that system for an applicant to receive admission in more than one university. This development led to the establishment of the Joint Admission and Matriculation Board in 1978 by the Federal Military Government. The establishment of the JAMB was a direct response to the failure of earlier system to effectively accommodate and guarantee the possibility of taking care of the growing aspirations of young school leavers seeking admission into the few universities in Nigeria. The responsibilities of the JAMB as specified by the law establishing the board were to conduct entrance examination and co-ordinate admission into various universities in such manner that an applicant would not deprive some other qualified candidates of places in the universities by being offered admission into two or more different institutions. The board has also offered admissions to reflect a fair distribution of students to universities to reflect the Federal character.

As the years roll by, more people have continued to raise alarm about the falling standard in education and criticized the quality of graduates being turned-out from Nigerian universities. Some employers of labour complain about the failure of graduates to meet their corporate expectations especially in terms of skills and competence. To

buttress this point, a study sponsored by the National Universities Commission and Education Tax Fund (NUC, 2004) found that lack of fit between university graduates and the needs of employers in various disciplines. The obvious lapses noticed in the process of admission of candidates into Nigerian universities through UME and the expected roles of universities in the society which is currently on the decline, necessitated the call for an alternative method of admitting students into Nigerian universities.

This call resulted in granting universities the power to conduct post-UME Universities conducted screening tests. It is believed by the proponents that post-UME-screening test will ensure quality and that when the best students are admitted, the results will also be enhanced. It is also believed that quality admission will produce better quality of graduates, students committed to their studies, reduced incidence of examination malpractice

2.5 Related Work

The quest for patterns in data has been studied for a long time in many fields, including statistics, patterns recognition and exploratory data analysis (Usama and Ramasamy, 2002). Analyzing data can provide further knowledge about a business by going beyond the data explicitly stored to derive knowledge about the business. This is where data mining has obvious benefits for any enterprise. Various algorithms and techniques like Classification, Clustering, Regression, Artificial Intelligence, Neural Networks, Association Rules, Decision Trees, Genetic Algorithm, Nearest Neighbour method are used for knowledge discovery from databases (Akinola *et al.,* 2012)

Naik and Ragothaman (2004) evaluated the ability of three different models neural networks, logit and probit to predict MBA student performance in graduate

programmes. In the work, neural network technique was used to classify applicants into successful and marginal student pools based on undergraduate Grade Point Average (GPA), Graduate Management Admission Test scores(GMAT), undergraduate major, age and other relevant data.

Three tables were used for the experiment carried out in the study and the results of the predictions of neural network and statistical models used showed that the neural network model outperformed the traditional statistical models and is a useful tool to predict MBA student performance. The overall results suggest that the classification accuracy (and implied predictive power) of the neural network model is 89.13 percent and that of the logit and probit models are 72.83 percent and 73.37 percent respectively. The use of a non-linear model such as the neural network allows administrators to specifically incorporate uncertainties into the decision making process. The traditional check list and formula approach do not permit this flexibility. To assess the neural network model's ability to classify students into successful and marginal groups, the predictive ability of the neural network model was compared with two statistical models namely logistic regression logit and probit. It was shown that the use of a neural network model can support and potentially improve decision making by MBA directors and deans.

Elden *et al* (2013) built a prediction model for predicting the student performance using a web-based tutoring system. This was done in order to improve the prediction results of the student academic performances of the work of Harb and Moustafa (2012) in which prediction was made using AdaBoost. In the previous work, feature selection techniques were applied to the dataset to select the most relevant features to reduce the size of the dataset. The experimental results show that all the classifiers give the best performance

with 3-7 features (out of sixteen). The proposed prediction model "Ada-GA" is implemented using the AdaBoost algorithm, with simple genetic algorithm, and it was found that the proposed "Ada-GA" classification model reduces the complexity of computation, while maintaining high detection accuracy compared to others. The prediction model predicts if the student will answer a certain problem true or false. With the reliable predictions of the performances of the students, the teachers can focus their efforts on those students who are likely failing in the final test, and thus helping them before difficulties become overwhelming.

Lubna (2009) carried out a study on implementing a technique that can be used to decide the best possible branch at the admission level which will guide the student towards the branch best suited for him. His work examines the entrance procedure for a student seeking admission to a college institute. The system "Artificial Student Decision (ASD)" was built and is a simple classifier system, which learns from the performance of the previous batches. This experience coupled with information about his aptitude enables the expert to guide the student towards the branch best suited for him. Genetics Based Machine Learning (GBML) system uses a Sample Aptitude Test (SAT), which tests a user (student) for different subjects, which are required to judge his aptitude. These subjects are stored in a database, which serves as an evaluating tool of the learning system, which uses it to decide the best possible branch at the admission level. These decisions are taken by the system based on certain ground rules. Apportionment of credits involved in the evaluation of aptitude is carried out using the famous Bucket Brigade Algorithm. The tripartite process of Genetic Algorithm has been applied to make the system robust. This work addressed an important issue in student education requirement, compares and contrasts what is involved in human learning with what is

involved in machine learning. The results shows in the long run for big knowledge based systems, learning will turn out to be more efficient than programming.

Wafa (2013) introduced a generic, multi-criteria model based on fuzzy logic concepts to build decision support system for admitting student into the university. He used Evolutionary computation techniques, fuzzy Logic, and genetic Algorithm. He firstly worked on a first order aggregation model and performed its learning stage by using Steady State Genetic Algorithms.

The main focus of his paper is to develop intelligent decision support system which spots the lights on the problems of multi-criteria decision that dealing with subjective and indefinite data.

Adeniyi et al (2010) used multiple regression analysis to analyzed the data of students admitted into the medical school via the JAMB, SSCE and Post- UME. He argued that only students who apply to read Medicine at the University and have scored a minimum of 200 UME qualify to take the Post UME test and that successful candidates at the Post UME (score $\geq$ 50%) would have their UME considered and those with the best UME scores are selected to fill various categories of admission . No student with Post UME score less than 50% got admission to read Medicine. His study was aimed at verifying the effectiveness of using the best UME score as the major criterion for admission. He attempted to determine the correlation between SSCE, UME, combination of both examination and academic performance of students in the first two years in the Medical school. Result showed that there was no correlation between UME score only and 100 level average score ($r = -0.054$, $P > 0.05$); but it correlates with 200 level results ($r = 0.318$, $P < 0.01$). There was correlation between SSCE and 100 level result ($r = 0.406$, $P < 0.001$) but not with 200 level average ($r = 0.176$, $P > 0.05$). CJSC correlates with 100

level weighted average ($r = 0.368$, $P < 0.001$) and 200 level result ($r = 0.240$, $P < 0.05$). Based on UME score only, 21.10% of students with UME score ≥ 240 as compared with 15.40% of those with UME score < 240 failed and were withdrawn at 100 level ($z = 0.695$, $P > 0.05$) . The performance of students with high derived SSCE ≥ 60 was significantly better than those with lower SSCE < 60 ($P < 0.01$). Result also showed that SSCE is the best predictor of student performance at 100 level. It is suggested that the quality of Secondary School Certificate result should be given good consideration in admitting students into Medical Schools.

Dimitris (2005) developed a predicting model using genetic algorithm based using induction of decision trees. He argued that students who enroll in the undergraduate program on informatics at the Hellenic Open University (HOU) demonstrate significant difficulties in advancing beyond the introductory courses. The Author embarked on analyzing the academic performance throughout the academic year, as measured by the homework assignments, and attempt to derive short rules that explain and predict success or failure in the final exams. This was because majority of students dropped out after failing to deliver the first one or two written assignments. He used decision tree representation based on fitness function. At each time-point (in genetic algorithms dialect: generation) a certain number of decision trees (population) is generated and sorted according to some criterion (fitness). Based on that ordering, certain transformations (genetic operators) are performed on some members of the population to produce a new population that will be subject to the same cycle until a predefined number of generations is reached (or no further improvement is detected). The approach was compared with others literature of Open University and argued that his concept has a potential for developing into an alert tool.

Adeyemi (2010) used descriptive analysis, to study academic performance of students in the three Universities in the two states, who offered Educational Management. The paper investigated the predictive ability of credit grades in Mathematics in senior secondary certificate (SSC) examinations in predicting the success of students in Educational Management in universities in Ekiti and Ondo States, Nigeria. Data were collected through an inventory and analyzed with the use of t-test, correlation analysis, analysis of variance and linear regression.

Testing the hypothesis using the t-test statistic, data on the performance of Educational Management students measured by the cumulative grade point average (CGPA) in Educational Management in the two universities were collected from the respondents using the inventory. The data were processed and subjected to statistical analysis. The findings in the t-calculated (.127) were less than the t-table (1.67). The hypothesis was accepted. This shows that there was no significant difference in the performance of students in Educational Management measured by the cumulative grade point average (CGPA) between the universities in Ekiti and Ondo States, Nigeria. This implies that the performance of students in Educational Management in the two universities was almost at the same level

Also, data on the number of Educational Management students in the two universities who scored credit and above grades in Mathematics in the senior secondary certificate examinations and the cumulative grade point average (CGPA were collected. Correlation analysis was computed while a correlation matrix was derived showing correlation coefficients for the pair of variables. The findings show the probability was less than 0.05. This shows that there was a significant relationship between the credit grade point in Mathematics obtained by Educational management students and the

cumulative grade point average (CGPA) in Educational Management in the universities. The correlation matrix shows the correlation coefficients between the pair of variables.

In order to determine the inter-correlation among variables, linear regression analysis was conducted. The predictor variable was the entry credit grade points in mathematics while the criterion variable cumulative grade point average (CGPA) in Educational Management in the universities.

In conclusion, the finding revealed that there was a significant relationship between the entry grade point of credit in Mathematics and the performance of Educational Management students measured by the CGPA in universities in the two States. It was also found that the entry credit grades obtained by Educational Management students in Mathematics in senior secondary certificate examinations on the findings of the study, it is recommended that more emphasis should be given to the teaching of mathematics in secondary schools in the two states to enable better performance of students in the universities.

Salahdeen and Murtala (2004), used descriptive statistic to determine the relationship between the quality of Senior Secondary School Certificate Examination results, Joint Admissions and Matriculation Board organized Universities Matriculation Examination (JAMB-UME) scores, and student's academic performance at Lagos state University College of Medicine (LASUCOM), Nigeria , records of students admitted into the Medical School in 1998/99 were studied. Data transformations were completed and subsequently standardized. Standard descriptive statistics and correlation coefficient were used to determine the correlation between the variables. The results showed that there was no significant correlation between the SSCE and the JAMB scores. In addition, correlation between JAMB and the student performance at 100 level and pre - clinical sciences also proved to be non-significant. These results indicate that SSCE is a

better predicator of students' performance at pre-clinical sciences examination than JAMB Scores.

Behrouz and William (2003) present an approach for classifying students in order to predict their final grade based on features extracted from logged data in an education web-based system using Genetic Algorithm. The research answers the question of finding classes of students (groups of students) who use the online resources in a similar way and if so, can the classes be identified. With the finding of similar patterns of use in the data gathered from LON-CAPA, one will be able to make predictions as to the most-beneficial course of studies for each learner based on their present usage. The system could then make suggestions to the learner as to how to best proceed. The objective was achieved by extracting features from the homework data and compares their performance on a dataset from LON-CAPA. As test data, they selected the student and course data of a LON-CAPA course, PHY183 (Physics for Scientists and Engineers I), which was held at MSU in spring semester 2002. This course integrated 12 homework sets including 184 problems, all of which are online. The grouping of the students regarding their final grades was handled in several ways as shown below:

1. Assign class labels to the possible 9 classes students' grades,
2. label the students in relation to their grades and group them into three classes, "high" representing grades from 3.5 to 4.0, "middle" representing grades from 2.5 to 3, and "low" representing grades less than 2.5.
3. categorize the students with one of two class labels: "Passed" for grades higher than 2.0, and "Failed" for grades less than or equal to 2.0.

Recombination of was handled using proportional selection and crossover operation was not performed on all strings in the population directly rather it was applied with a probability P_x when the pairs are chosen for breeding with probabilty $P_x = 0.7$.

Offspring = parent1 + Alpha × (parent2 – parent1) (3) where Alpha is a scaling factor chosen uniformly in the interval [-0.25, 1.25]

A further genetic operator, mutation is applied to the new chromosomes, with a set probability P_m. Mutation causes the individual genetic representation to be changed according to some probabilistic rule.

Mutated Var = Var + MutMx × range × MutOpt(2) × delta where delta is an internal matrix which specifies the normalized mutation step size; MutMx is an internal mask table and MutOpt specifies the mutation rate and its shrinkage during the run. The mutation operator mutbga is able to generate most points in the hypercube defined by the variables of the individual and the range of the mutation. MATLAB was used for the implementation of the GA to optimize classification performance.

2.6 Literature Gap

From the available literature, it is noted that different analysis has been done on students' results repository, which qualifies them for admission into universities. However, the aggregate in all their various examination results that qualifies them for admission has never been analyzed for hidden but important patterns. Analyzing and searching for these patterns with respect to their performance in the first academic year could be of a great importance to academic planners in enhancing their decision making process and improving student performance. In order to bridge this gap, this thesis presents an analysis of aggregate of UME, PUME, O'level and first year CGPA in predicting the best pattern that are mostly occurrence in the repository of students records, using genetic algorithm. This is to discover the hidden relationships that exist between different students admission combinations in form of rules. The generated rules are analyzed to make useful and constructive recommendations to the academic planners. This will enhance academic planner's sense of decision making and aid in the

curriculum structure and modification which in turn improve students' performance and trim down failure rate.

CHAPTER THREE

MODELING THE STUDENT'S PERFORMANCE DECISION SUPPORT SYSTEM

3.1. System Description

A predictive model which determines the relationships between the entry requirement and the first year CGPA is developed. The admission dataset that represents the entry results of student previously admitted into the programme were first normalized and their features extracted. A list of classifiers was generated for each student performance and grouped according to their respective CGPA. Genetic algorithm is then used as an optimization for searching meaningful relationship between the classifiers by means of reproduction till the best combination is found. Figure 3.1 shows the relationship between the system modules.

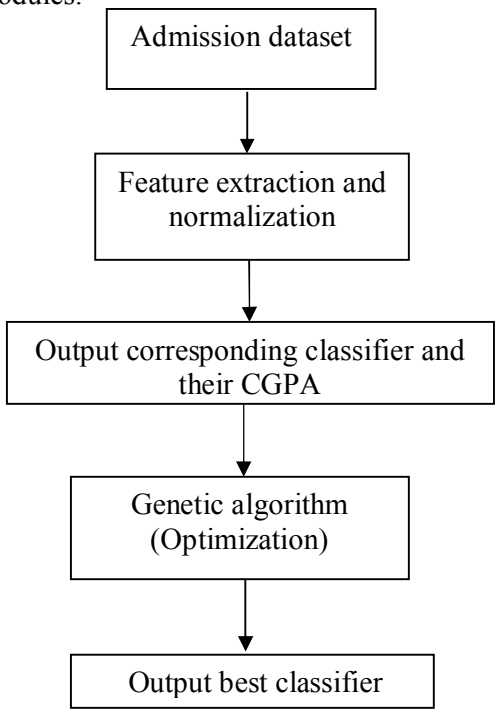


Figure 3.1: System modules relationship

3.2. Benchmarking Student's Performance

Let P be the programme a candidate can apply for; $E = \{E_1, E_2, \ldots, E_j\}$ be the set of examinations defines for admission into any of the programme and let $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_j\}$be the set of weight outlined to those examinations based on their qualities in predictive measure such that $\alpha_j \leq \alpha_{j-1} \leq \ldots \leq \alpha_1$ and $\alpha_1+\alpha_2+ \ldots+ \alpha_j = 1$. For instance, if E_1 and E_2 as examinations and also α_1E_1 and α_2E_2 such that $\alpha_1>\alpha_2$ then examination E_1 is more important than examination E_2. Let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of accepted subjects for admission into programme P, a candidate must have taken some of these subjects during the examination E, let G be the notation for grade equivalent in these subjects. Then a representation for candidate i in examination E is define as $E_i = \{s_{1G}, s_{2G}, \ldots, s_{nG}\}$. The representation for a candidate's performance in E examination for all the S subjects taken in such an examination can then be represented by column vector with attribute α which indicates the weight for such an examination.

$$E_{ij} = \alpha_j \begin{pmatrix} s_{1G} \\ s_{2G} \\ \ldots\ldots \\ s_{nG} \end{pmatrix} \ldots \quad \ldots \quad \ldots \quad 3.1$$

Consequently, for j number of examinations taken by the candidate i, it can be represented by:

Subject	Exam 1	Exam 2	…	Exam j
$\begin{pmatrix} subject1 \\ subject2 \\ \ldots\ldots\ldots\ldots \\ subjectn \end{pmatrix}$	$\alpha_1 \begin{pmatrix} s_{1G} \\ s_{2G} \\ \ldots\ldots \\ s_{nG} \end{pmatrix}$	$\alpha_2 \begin{pmatrix} s_{1G} \\ s_{2G} \\ \ldots\ldots \\ s_{nG} \end{pmatrix}$	...	$\alpha_j \begin{pmatrix} s_{1G} \\ s_{2G} \\ \ldots\ldots \\ s_{nG} \end{pmatrix}$

Figure 3.2: Representation of student's subject in each examinations

3.3 Features Extraction and Normalization of Data

The variables declaration and parameters establishment are worked out on the admission dataset as discussed in the benchmark in section 3.2. The admission dataset is then subjected to feature extraction and normalization. Feature extraction is carried out since the data collected can be inconsistence, incomplete or noisy. This may be as a result of number of factors ranging from data entry or transmission problem, discrepancy in the naming convention, duplicated records or incomplete data or removal unwanted entries that are not required. The normalization of each examination grading system is carried out on the exams involved to bring a unique grade pattern.

3.3.1 Handling Discrepancy of Feature Extraction

Case 1: In some examination, a candidate may be given the privilege to combine results for such an examination for certain number of sittings. The candidate may not have sat for all the subjects in all the sittings. In that case there is need to reconcile the entry in order to have a unique entry for each subject in each examination that reflects the real performance of the student.

For any examination in which more than one sitting is used, the average score in the sittings should be calculated such that

$$S_{iG} = \sum_{i=1}^{n} s_{i}/n \qquad \ldots \qquad \ldots \qquad \ldots \qquad 3.2,$$

where n is the number of sittings allowed.

Case 2: In computing the candidate's performance in all the examinations taken, a lot of problems maybe encountered such as the case where subject have empty entries. There are two possibilities to address the matter.

1. The matrix representation of a particular subject has at least half entries. This implies that the candidate has taken more than half number of same subject in different examinations. For such a case, any empty entry can be filled by computing the average grade of the student from other examinations for that subject and the obtained value is used to replace the missing value.
2. The matrix representation of a particular subject has less than half entries. This implies that the student has taken a particular subject fewer times. Because of this we cannot really evaluate the performance of the candidate in such subject. In this case, such subjects are totally ignored and consequently, the few values appearing in the matrix representation of the subject are deleted.

Example One

Case 1: In the case where a candidate combines his O'level results

Sitting 1

Subject	WAEC
$\begin{pmatrix} English \\ Mathematics \\ Physics \\ Chemistry \\ Bio\log y \end{pmatrix}$	$\begin{pmatrix} C_6 \\ B_3 \\ C_6 \\ B_3 \\ C_4 \end{pmatrix}$

Sitting 2

Subject	WAEC
English	*F*9
Mathematics	*C*5
Physics	*D*7
Chemistry	*B*3
Bio log *y*	*A*1

The performance is converted to equivalent points

Subject	WAEC	WAEC
English	3	0
Mathematics	4	3
Physics	3	2
Chemistry	4	4
Bio log *y*	3	5

The average performance of the candidate in each subject is being determined as:

Subject	O'level
English	1.5
Mathematics	3.5
Physics	2.5
Chemistry	4
Bio log *y*	4

Case 2:

1. For a subject with empty entries where the candidate has taken more than half the number of same subject in different examinations

Subject	O'level	UTME	Post-UTME
English	C_6	45	67
Mathematics	B_3	67	34
Physics	–	56	45
Chemistry	B_3	20	32
Bio log *y*	C_4	78	56

The equivalent point is assigned to the grade system of each examination

Subject	O'level	UTME	Post-UTME
English	3	2	4
Mathematics	4	4	0
Physics	–	3	1
Chemistry	4	0	0
Bio log *y*	3	5	3

The empty entry is filled by finding the average of the subject in the other examinations.

Subject	O'level	UTME	Post-UTME
English	3	2	4
Mathematics	4	4	0
Physics	2	3	1
Chemistry	4	0	0
Bio log *y*	3	5	3

i. For a subject with empty entries and the matrix representation for the particular subject has less than half of the entries.

Subject	O'level	UTME	Post-UTME
English	C_6	45	67
Mathematics	B_3	67	34
Physics	–	56	–
Chemistry	B_3	20	32
Bio log *y*	C_4	78	56

Here, the physics entry is deleted from the student result

Subject	O'level	UTME	Post-UTME
English	C_6	45	67
Mathematics	B_3	67	34
Chemistry	B_3	20	32
Bio log *y*	C_4	78	56

3.3.2 Handling of Normalization

Normalization of data is carried out on the subject grades so as to have uniform data representation for various predictive variables of the entrance result. Scores are normally represented in two class categories called grade descriptors. The first grade descriptor uses (A1,A2,B3,C4,C5,C6,…,) and the second grade descriptor uses (A,B,C,D,E,F) . For instance SSCE result is represented using secondary grade descriptors while university grading system uses primary grade descriptor. The grading system of examination such as JAMB does not fall into any of these categories. All these discrepancies must be harmonized to obtain a single representation for all results. The essence of this is to have results that are uniform and easy to work on by the genetic operators. The representation should be in such a way that the result converted from a different descriptor into newly formulated one still preserve the real performance as it was in the previous one.

In this thesis, the primary grade descriptor is used as a standard for the conversion and is assigned the necessary equivalent points as shown in table 3.1.

Table 3.1: Grade conversion table

Class range (%)	**Grade descriptors**	**Notation(primary grade descriptors)**	**Points**
70 and above	A,A1	A	5
60 – 69	A2-A3,B,B2-B3	B	4
50 – 59	C,C4-C6	C	3
40 – 49	D	D	2
0 – 39	E	E	1

Since the form of examinations that are majorly going to be working on are O'level, UTME and Post-UTME, the categories in which each of this examination falls into is noted, which could be of class range(%) and grade descriptors. An attempt is made to convert them into our own notation and an appropriate equivalent point of this notation is assigned to them.

To determine a candidate's performance in all the examinations taken, an appropriate weight is assigned to each examination according to their level of importance and an aggregated value is obtained in all the subjects taken.

The performance of a candidate applying for a programme p can be computed as follows:

$$\alpha_1 \begin{pmatrix} s_{1G} \\ s_{2G} \\ \cdots \\ s_{nG} \end{pmatrix} + \alpha_2 \begin{pmatrix} s_{1G} \\ s_{2G} \\ \cdots \\ s_{nG} \end{pmatrix} + \ldots + \alpha_j \begin{pmatrix} s_{1G} \\ s_{2G} \\ \cdots \\ s_{nG} \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}$$

Consequently, $x_m = \sum_{i=1}^{j} \alpha_i s_m$ 3.3,

where $1 \leq m \leq n$

Example Two

For a candidate C, who sat for UTME, Post _UTME and has O'level results as required by the department, it is possible to calculate the average performance for such a candidate as follows:

Subject	O'level	Post UTME	UTME
English	C_6	45	67
Mathematics	B_3	67	34
Physics	C_6	56	45
Chemistry	B_3	20	32
Bio log *y*	C_4	78	56

Assuming the levels of importance of the examinations O'level, PUTME, UTME, are in these order and the weight 0.5, 0.3, and 0.2 is assigned respectively. Then using the grade points as in the table 3.1, we can represent the result as follows:

Subject		O'level		Post UTME		UTME
English		C_6		45		67
Mathematics		B_3		67		34
Physics	= 0.5	C_6	+ 0.3	56	+0.2	45
Chemistry		B_3		20		32
Bio log *y*		C_4		78		56

To normalize we use the grade points as in table 3.1, and the result is represented as follow:

$$\begin{pmatrix} English \\ Mathematics \\ Physics \\ Chemistry \\ Bio\log y \end{pmatrix} = 0.5 \begin{pmatrix} 3 \\ 4 \\ 3 \\ 4 \\ 3 \end{pmatrix} + 0.3 \begin{pmatrix} 2 \\ 4 \\ 3 \\ 1 \\ 5 \end{pmatrix} + 0.2 \begin{pmatrix} 4 \\ 1 \\ 1 \\ 1 \\ 3 \end{pmatrix}$$

By multiplying the points with the coefficient we obtain

$$\begin{pmatrix} English \\ Mathematics \\ Physics \\ Chemistry \\ Bio\log y \end{pmatrix} \begin{pmatrix} 0.5*3+0.3*2+0.2*4 \\ 0.5*4+0.3*4+0.2*1 \\ 0.5*3+0.3*3+0.2*1 \\ 0.5*4+0.3*1+0.2*1 \\ 0.5*3+0.3*5+0.2*3 \end{pmatrix}$$

$$\begin{pmatrix} English \\ Mathematics \\ Physics \\ Chemistry \\ Bio\log y \end{pmatrix} \begin{pmatrix} 2.9 \\ 3.4 \\ 2.6 \\ 2.5 \\ 3.6 \end{pmatrix} = \begin{pmatrix} S_{1G} \\ S_{2G} \\ S_{3G} \\ S_{4G} \\ S_{5G} \end{pmatrix}$$

Where $S_{1G,}$ $S_{2G,}$...,S_{5G} are variables representing the corresponding subjects grade respectively

3.4 Genetic Algorithm

Genetic algorithm (GA) is an evolutionary based stochastic optimization algorithm with a global search potential proposed by Holland (1973). GA maintains a population pool of candidates and is among the most successful class of algorithms under Evolutionary Algorithms (EAs) which is inspired by the evolutionary ideas of natural selection. They follow the principles of Charles Darwin Theory of survival of the fittest. However, because of its outstanding performance with optimization, GA has been regarded as a function optimizer.

The algorithm begins by initializing a population of solution (chromosome) and comprises representation of the problem usually in the form of a bit vector. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated, using some measures of fitness (using an appropriate fitness function suitable for the problem). To create the next generation, new chromosomes, called offspring, are formed by either merging two chromosomes from the current generation using a crossover operator or modifying a chromosome using a mutation operator. A new generation is formed by selecting fitter chromosomes that have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which represents the optimum or a suboptimal solution to the problem. The three principal genetic operators in GA involve selection, crossover, and mutation (Binitha and Sathya, 2012). Figure 3.3 shows the outline of the GA algorithm

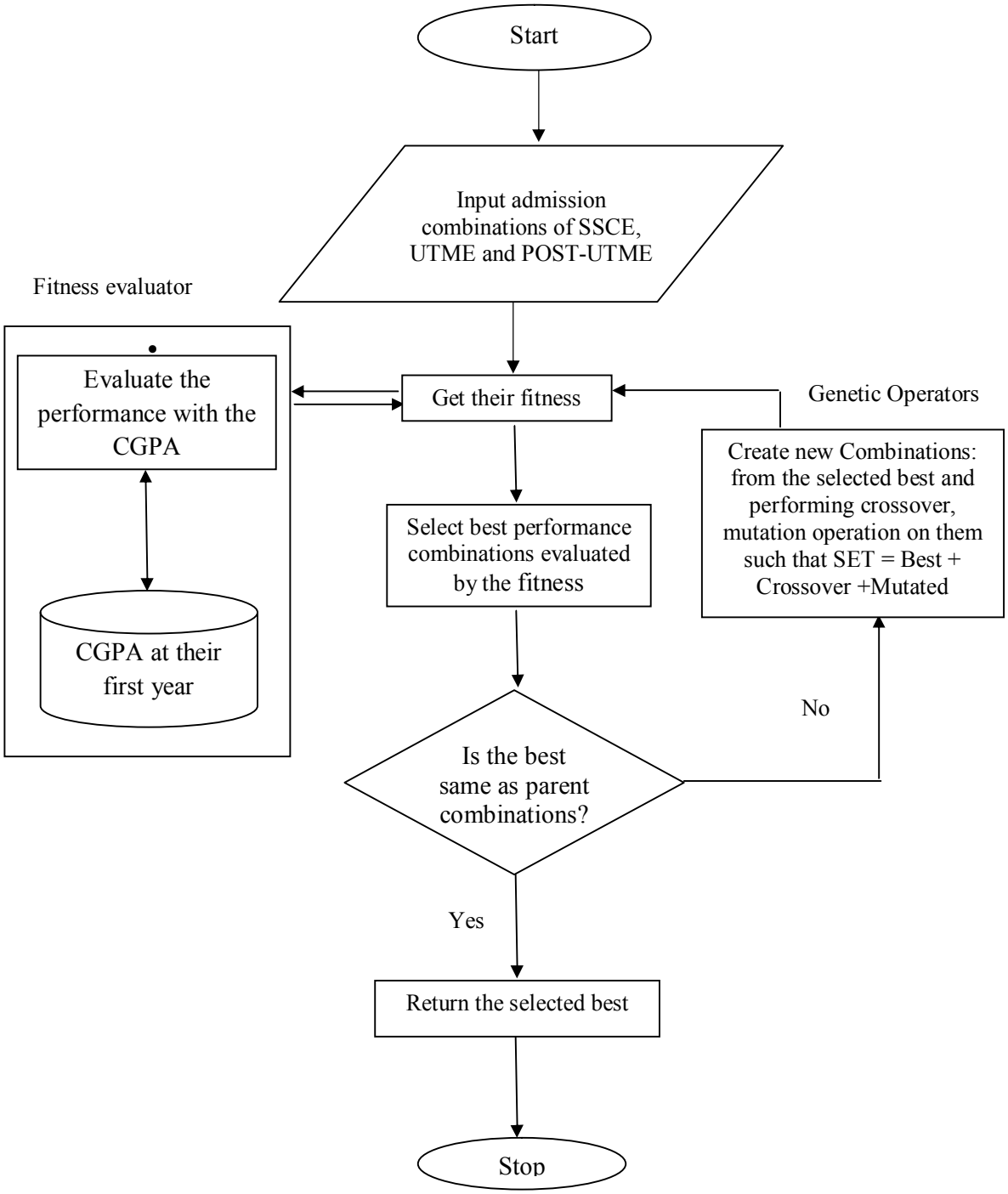


Figure 3.3: Genetic Algorithm flowchart: Adapted from (Sanjay et al., 2011)

3.4.1. Fitness Measurement

Fitness measurement gives a criterion to rank candidate solutions which serve as the basis of making a decision as to whether a particular individual solution is fit to survive and reproduce. It is of important to assign priority to each classifier according to the performance of students in their 100 level CGPA.

In determining the fitness of rule classifier the following procedure must be followed using (Behrouz and William, 2003) CGPA grouping:

1. Assign possible labels to each of the unique rule classifier
2. Group the students CGPA of the classifier into three classes "Good" "Average" and "Poor" as shown in table 3.2.

Table 3.2: Grouping of CGPA into different classes

Performance (CGPA)	**Interpretation**
CGPA ≥ 3.5	Good
$2.0 <$ CGPA< 3.5	Average
CGPA ≤ 2.0	Poor

3. For each unique classifier, count the number of those that falls under each group
4. Calculate the unit points of each classifier
5. Determine the fitness of each unique classifier by using equation 3.4 and store the result into an appropriate rule profile
6. Sort the classifier based on their fitness in descending order

$$fitness = \begin{cases} \dfrac{number_good + number_average}{number_average + number_poor} * unit_point \; .....if \\ average + poor > 0 \end{cases} \quad \ldots \quad \ldots \quad \ldots \qquad 3.4$$

$$number_good * unit_point \; if$$
$$average + poor = 0$$

3.4.2 Selection

Candidates for mating are chosen by a procedure of natural selection. This can be described by using proportional selection. According to this procedure, a calculation of g(x) is generated for each classifier, where g(x) is the relative fitness defines by equation 3.5 as explained by (Magnus, 2003).

$$g(x) = \frac{\sum f(x)}{f(x)} \ldots \qquad \ldots \qquad \ldots \qquad 3.5$$

Where f(x) is the fitness for each classifier and ∑f(x) is the summation of all the fitness of the classifier.

In general, the purpose of this procedure is to eliminate chromosomes with low fitness and duplicate those with high fitness and by this the best is chosen and duplicated.

3.4.3 Crossover

After pairing selection, crossover operator perform task on the selected pairs of classifier. The procedure for crossover is in two steps:

1. Two classifiers are selected
2. Split classifiers at the crossover points using crossover probability (P_c)
3. Create children by exchanging tails
4. Probability of crossover is typically in the range (0.6 ,0.9).

Here the uniform point crossover point is used.

3.4.4 Mutation

Here, the selected pairs for mating are mutated based on certain probability P_m. The mutation mimics the phenomenon of natural mutation.

1. Alter each gene independently
2. Mutation probability is typically in the range (1/population_size, 1/chromosome_length)

3.5 Basic Genetic Algorithm Procedure

3.5.1 Initial Population Generation

The admission dataset of previous students admitted into the programme is collected. The feature extraction and normalization of data was carried out on the dataset to produce dataset that is in a format suitable for the application. As part of normalization and extraction process, unwanted column are discarded and appropriate weight are assigned to each examination category in order to generate each student subject pattern. The candidates average performance in all the three exams for the set of classifier rules. The initial population of the classifiers are denoted by I_k where k=1 to n and a random generation of m is selected from the classifier list. The value of m is important because if it is too large the algorithm will not be differing much from an exhaustive search and if too small, the algorithm may not reach the optimal solution. For illustration, table 3.3 shows the classifier rules for each student's performance and their corresponding CGPA classification.

Table 3.3: An illustrating example of students' classifier rules with their CGPA classes

Classifier	CGPA	Classification
DDDCD	1.34	Poor
ECDDC	2.00	Poor
EBCDA	2.78	Average
ECDDC	1.87	Poor
BADEA	3.68	Good
EBCDA	3.34	Average
EBCDA	1.40	Poor
EBCDA	1.65	Poor
EBCDA	1.20	Poor
CBADC	4.50	Good
CBACC	3.87	Good

CBADE	1.15	Poor
CBADE	2.73	Average
ABEDA	4.67	Good
ECDDC	1.21	Poor
ECDDC	1.45	Poor
DCDDC	1.89	Poor
ECDDC	1.67	Poor
CBADC	4.15	Good
ABEDA	2.38	Average
ABEDA	3.00	Average
CBADC	3.84	Good
CBADC	2.56	Average
BADEA	3.48	Average
BADEA	2.12	Average
ABEDA	1.45	Poor
ABEDA	1.87	Poor
BADEA	2.34	Average
BADEA	4.80	Good
CBACC	4.11	Good
CBADE	4.00	Good
CBACC	2.65	Average
ECDDC	3.45	Average
EBCDA	1.32	Poor
ECDDC	1.46	Poor
CBACC	2.57	Average
CBADE	1.88	Poor
CBACC	3.98	Good
BADEA	3.56	Good
CBADE	1.56	Poor
CBACC	3.11	Average
CBADE	2.34	Average
CBACC	4.16	Good
CBADC	1.76	Poor

3.5.2 Fitness Evaluation

Take for instance, suppose the initial populations selected are ECDDC, EBCDA, CBADC, ABEDA, BADEA, CBADE and CBACC as shown in table 3.3. Table 3.4 and 3.5 give the fitness value for the classifiers. The tables were generated using equation 3.4.

Table 3.4: CGPA grouping and the unit points

Class label	Rule set (classifier)	Unit Points	Number of outcomes	Good	Average	Poor
1	ECDDC	10	7	0	1	6
2	EBCDA	14	6	0	2	4
3	CBADC	17	5	3	1	1
4	ABEDA	17	5	2	2	1
5	BADEA	17	6	3	3	0
6	CBADE	15	6	1	2	3
7	CBACC	18	7	4	3	0

Table 3.5: Fitness table

Class label	Rule set	Fitness
1	ECDDC	1.42
2	EBCDA	4.67
3	CBADC	28
4	ABEDA	22.67
5	BADEA	28
6	CBADE	9
7	CBACC	42

3.5.3. New Population

The process of genetic algorithm involves repeating the steps for several iterations, until there is no difference between best outcomes and the previous parent combinations. The

creation of a new population is repeated by following steps below until the termination condition is satisfied.

Step 1: Selection

Following the fitness table in table 3.5 then the selection g(x) for each classifier is calculated using equation 3.5 above.

Table 3.6: Illustration of the proportional selection method

Class label	**Rule set**	**Fitness**	**Selection g(x)**
1	ECDDC	1.42	0.010
2	EBCDA	4.67	0.034
3	CBADC	28	0.206
4	ABEDA	22.67	0.167
5	BADEA	28	0.206
6	CBADE	9	0.066
7	CBACC	42	0.309

The best combinations selected for crossover and mutation are shown in Table 3.7

Table 3.7: The best combinations selected for mating

Rule set	**Class label**
CBADC	3
BADEA	5
ABEDA	4
CBACC	7

Step 2: Crossover

Crossover is perform on the parents to form new offsprings. The best combinations selected for mating can pair up as shown in Table 3.8 below

Table 3.8: pairing classifiers for crossover

Class label	Rule sets
7	CBACC
3	CBADC
7	CBACC
5	BADEA
5	BADEA
3	CBADC
7	CBACC
4	ABEDA
3	CBADC
4	ABEDA
5	BADEA
4	ABEDA

Let pc=0.7. Table 3.9 shows the result of the crossover where the cross over probability is 0.7. The reason for the choice of the value is that the crossover probability P_c, controls the rate at which solutions are subjected to crossover and the higher the value, the quicker are the new solutions introduced into the population. Typical values of P_c are in the range 0.6 -0.9.

The random number are defined by using the java random function rand, as follow:

Random number [i]= rand(1, 99)/100 where 1<i<5

Table 3.9: crossover to form new rules

Parent 1	C	B	A	C	C
Parent 2	C	B	B	D	C

Random number	0.89	0.32	0.76	0.34	0.36
Crossover probability=0.7	>0.7	<0.7	>0.7	<0.7	<0.7

Offspring 1	C	B	B	C	C
Offspring 2	C	B	A	D	C

The process is repeated for the selected classifiers.

Step 3: Mutation

Mutation involves the modification of the value of each 'gene' of a solution with some probability P_m, (the mutation probability). The role of mutation in GA has been that of restoring lost or unexplored genetic material into the population to prevent the premature convergence of the GA to suboptimal solutions.

Nevertheless the choice of p, is critical to GA performance. Large values of p, transform the GA into a purely random search algorithm, while some mutation is required to prevent the

premature convergence of the GA to suboptimal solutions. Typically p, is chosen in the range (1/population size, 1/ chromosome length). Assume that the probability of the mutation is 0.05.

by randomly taking a position from each offspring as shown in table 3.10.

Table 3.10: mutated classifiers

Rule set	**Mutated rule set**
CBA[D]C	CBACC
BA[D]EA	BACBA
ABE[D]A	ABECA
C[B]ACC	CAABC
[C]BAEA	BBAEA
BADC[C]	BADCB
BAA[D]C	BAACC
BB[E]DA	BBDDA
AA[D]EA	AACEA
CBD[E]A	CBDDA
CB[E]DA	CBDDA
A[B]ACC	AAACC
ABAD[C]	ABADB
BBE[D]A	BBECA

3.5.4 Acceptance

The SET = (best □ crossover □ mutated) is tested with the termination criteria. If the condition is satisfied it is accepted and if is not, the new population is assumed to be the SET = (best □ crossover □ mutated) of the immediate level and the step of fitness, selection, crossover and mutation as discussed above are repeated.

3.5.5 Termination Condition

The generational process of GA is repeated until a termination condition has been reached. The termination conditions are:

1. The highest ranking solution‘s fitness has reached a plateau such that successive iterations no longer produce better results.

   The criterion to stop the genetic iteration is by comparing the new population with the immediate initial population. If the evaluation of the fitness of the generated SET = best □ crossover □ mutated of classifiers is the same as the preceding best population then the iteration is to stop otherwise it continues till the desire is obtained. When the stopping condition is meet, the SET = best □ crossover □ mutated is outputted. If the termination condition is not satisfied, the loop is repeated.

2. Fixed number of iteration is reached without converging.

   To avoid infinite loop, if after n number of iteration where n is defined by the user the loop stop. This is done so that the best combinations will not take too time or in the case where the best combination may not be found based on the input.

CHAPTER FOUR

IMPLEMENTATION OF THE DECISION SUPPORT SYSTEM

4.1 Introduction

This chapter presents the detailed implementation of the proposed system using the concept of Genetic Algorithm. The students' admission data set for computer science students, mathematics department of Ahmadu Bello University, Zaria was used in evaluation of the system and the actual implementation is presented in this chapter.

The system was implemented using Java. Java is known to be platform independent, that is, it can be deployed on different platforms (like Microsoft windows, Unix, Mac OS, Android). It has rich libraries to perform any task. It also makes possible to use java classes to handle tables as required by the work.

4.2 System Requirement

4.2.1 Hardware Requirement

The evaluation test was performed using Processor 2.40GHz, RAM 4GB and Windows 8 on the standalone PC.

4.2.2 Software Requirement

The system for the application runs on windows 8, and it was developed using Netbeans IDE environment version 7.1, Mysql database was used to store the data. Since, the system was developed using java, Java Development Kit (JDK), was installed.

4.3 Data Handling

4.3.1 The Data Use for Evaluation

The idea of this thesis is to build admission decision support system for mining trends of pattern from a large volume of admission datasets from Computer science, Mathematics Department, Ahmadu Bello University. The decision support system (dss)

is an application that takes in dataset of admission students' details which are UTME, PUTME, and O'level, respectively and their associated performance at the end of their first year in the programme. The data is prepared in a Microsoft excel format, from which it can be uploaded into mysql database.

4.3.2 Data Normalization

The data set collected from the department are large, some of which are not relevant for the study. It is therefore very important to normalize the data. In the course of normalization, data like, aggregate score in each examination, candidates' name and their state of origin were all removed. Student scores in the respective subject of UTME, PUTME, and O'level examination were converted to grade points. As part of the normalization process, the performance of students' in the important subjects, were average out bearing in mind the importance attached to each examinations as explained in chapter three.

4.3.3 Data Storage

The goal of this work is to mine for a pattern, thus, there is need for a data repository. The data set collected from the department were in different format (excel or Microsoft word), for the usage of this application, all the data were stored in excel format, from this format, the system is designed to be able to upload an excel file and stored it in Mysql database.

4.4 System Implementation

The system was implemented in four units: the upload unit, which handles the procedure for uploading of admission data, the second unit enables selection of subjects on which searching is to be performed, while the third and fourth units performs the searching techniques using the concept of genetic algorithm.

4.4.1 Upload Unit

This unit takes admission data based on the faculty and the department of choice using methods defined in them. Figure 4.1 to 4.4 shows the interface. A dialog box is available for faculty and department selection and the year of files to be uploaded. The method used in faculty, department and files upload is discussed as follow:

Whenever an upload button is selected, a constraint checks whether faculty, department and year are selected, if any of these is not selected, a message is popped up prompting the user to select whichever requirement that has not been selected. At the selection, the constraint check the file extension matching it with the excel format. At a successful matching, a message popped up showing that the file is uploaded successfully. The system is built to be able to take as many files as possible, after the first upload, therefore, the system prompt the user for more uploads, if any.

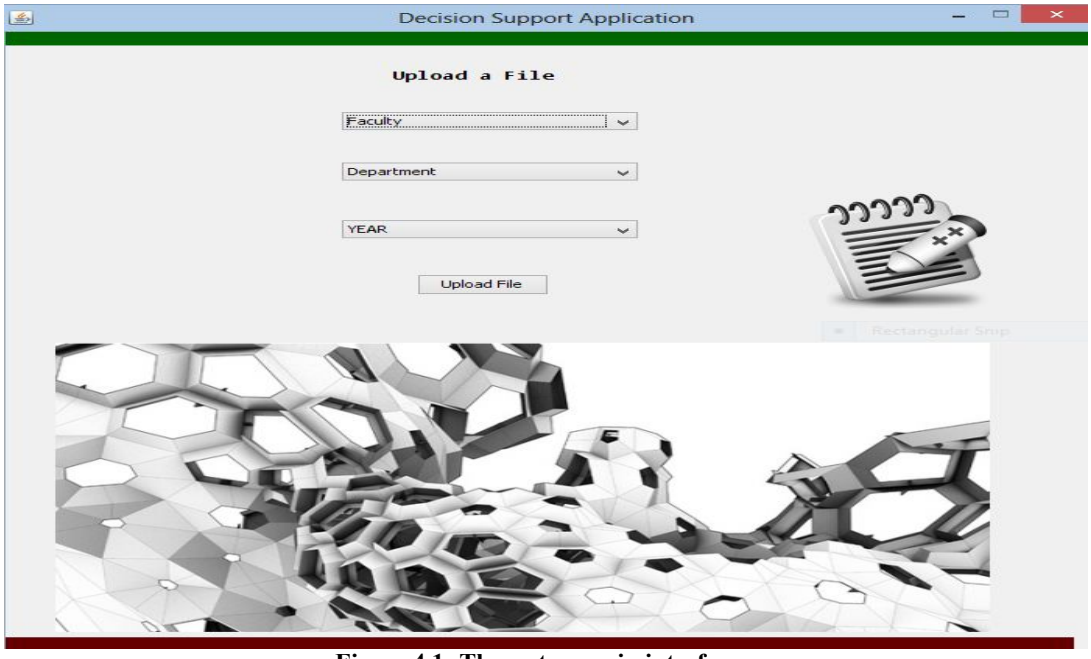


Figure 4.1: The system main interface

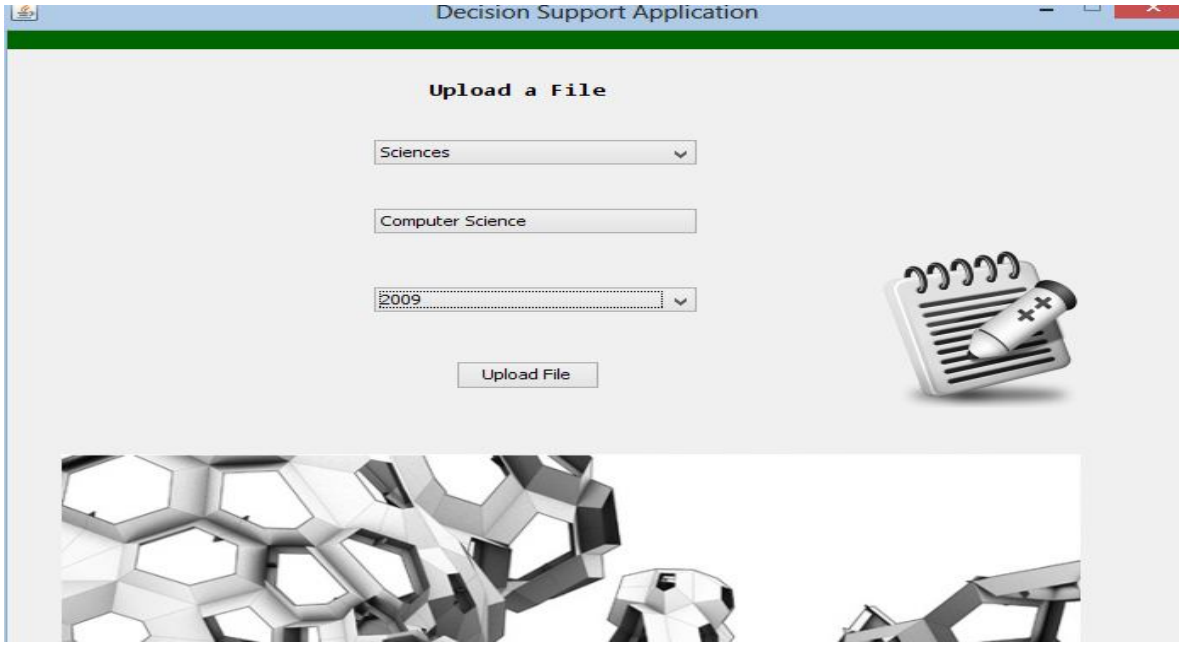


Figure 4.2: Faculty, department and the session of year of data selection

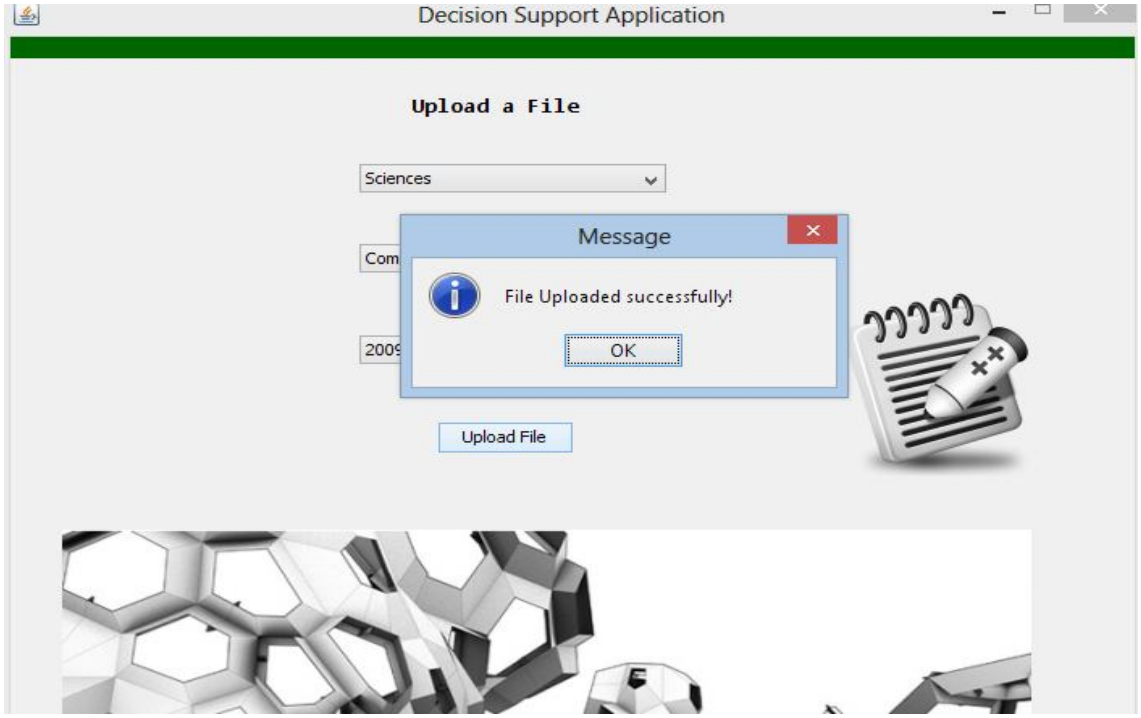


Figure 4.3: Report of successful upload of file

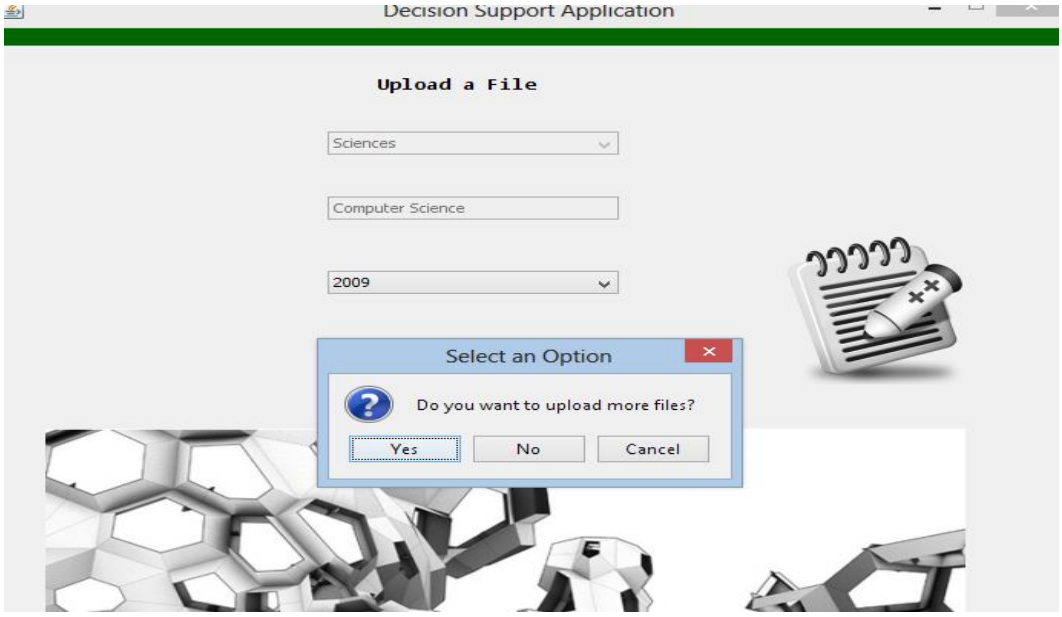


Figure 4.4: Interface prompting user for more upload

4.4.2 Subjects Selection Unit

The second unit has an interface where the administrator selects the subject combinations for the UTME. The administrator indicates the subjects where he/she wants to confine his searching based on department requirement. Fig 4.5 shows the interface.

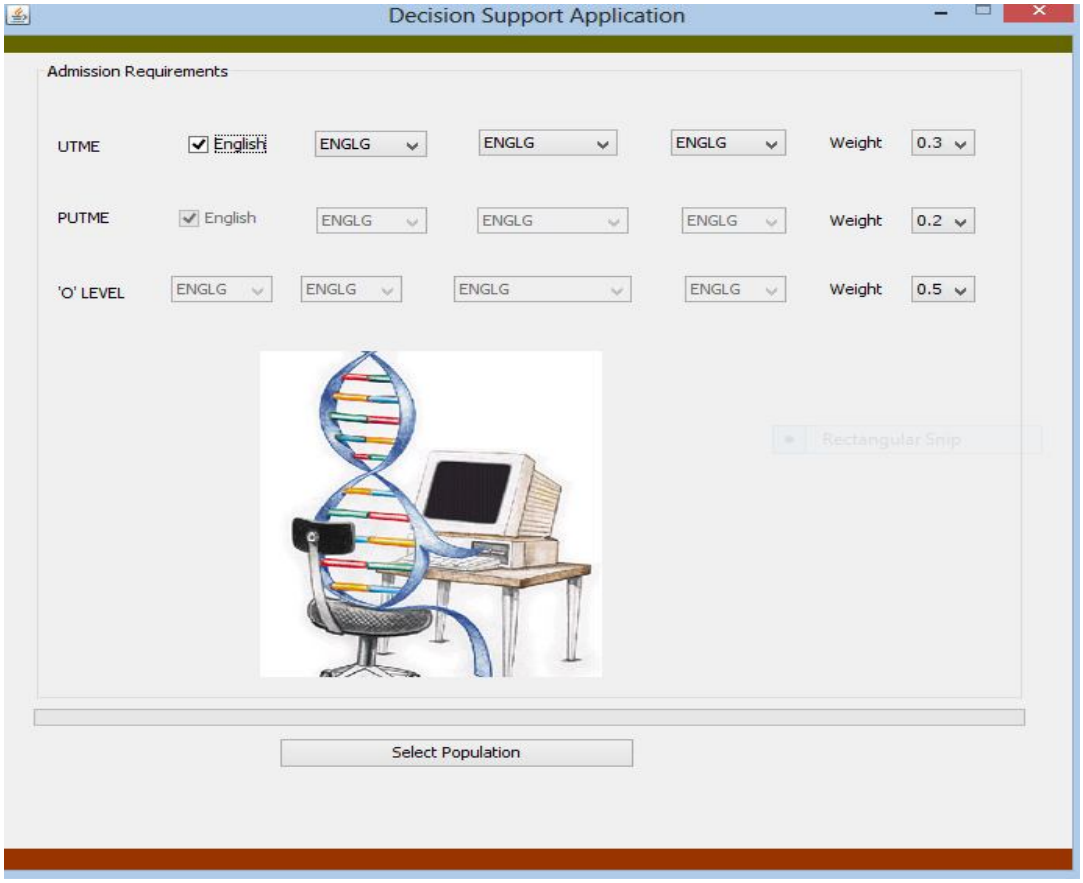


Figure 4.5: System interface for subject selection

The selection of UTME subject combinations, result to automatic selection of the same subjects for the PUTME and O' levels. Appropriate weight for UTME, PUTME and O'level is also indicated. The system searches for the subject combinations from the uploaded excel file and categorize them into UTME, PUTME, O'level and CGPA result tables respectively. The system generates pattern for each subject combination using the score provided from the tables (UTME, PUTME, O'level) and multiplying the appropriate examination weight to compute the average for each subjects.

4.4.3 Selection Unit

This is the unit that starts the genetic algorithm (selection) as shown in figure 4.6. The interface has a text field where the user enters the number of selection or population size of interest. This is a command to the system to select certain number of best

combinations based on fitness that has been calculated. This is actually the initial population size.

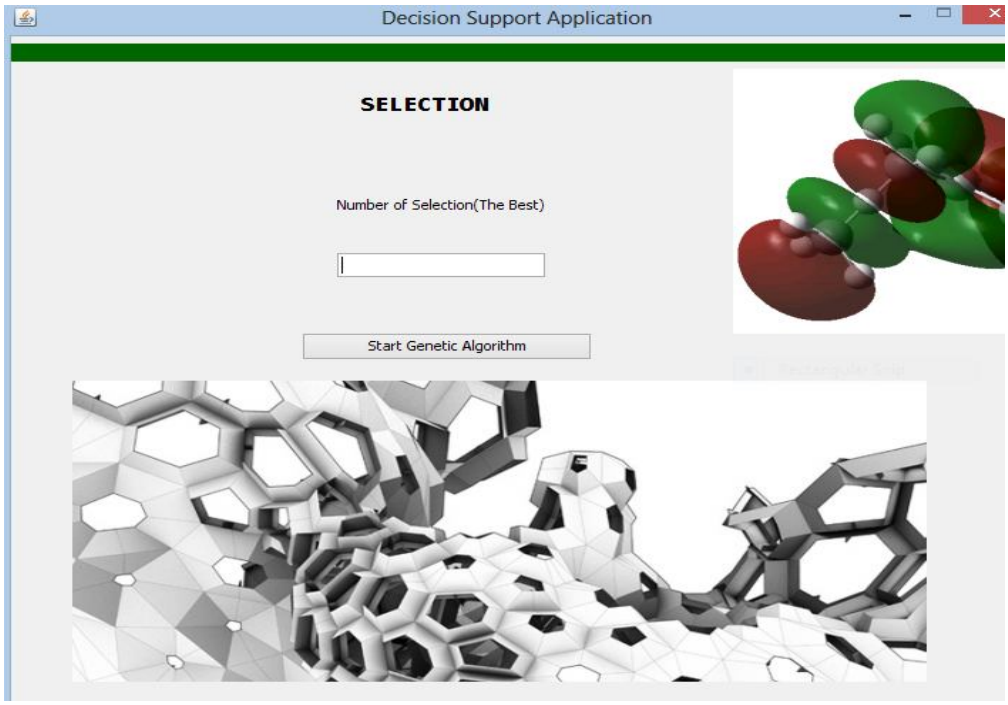


Figure 4.6: System interface for population size

4.4.4 Searching Unit

The forth unit provide a platform where the searching of best pattern takes place. The interface is shown in Figure 4.7. The platform is divided into three sections:

1. The log text field,
2. The options (JAMB, PUTME, O'level, 100 level, relative and fitness table respectively and also graph, file uploaded) and
3. The best pattern text field.

The log text area shows the progress of the genetic algorithm. It shows the result of selection, combination of the selected patterns, crossover, mutation, parent population, children population and the number of iteration performed.

Below this log text area is best pattern text field that shows the result of the best patterns over a certain number of iterations.

The graph button shows the graphical representation (bar chart) of the best pattern of the data mined as display on the text field of best pattern. And the file uploaded button shows the file uploaded for the system to mine and perform genetic algorithm on.

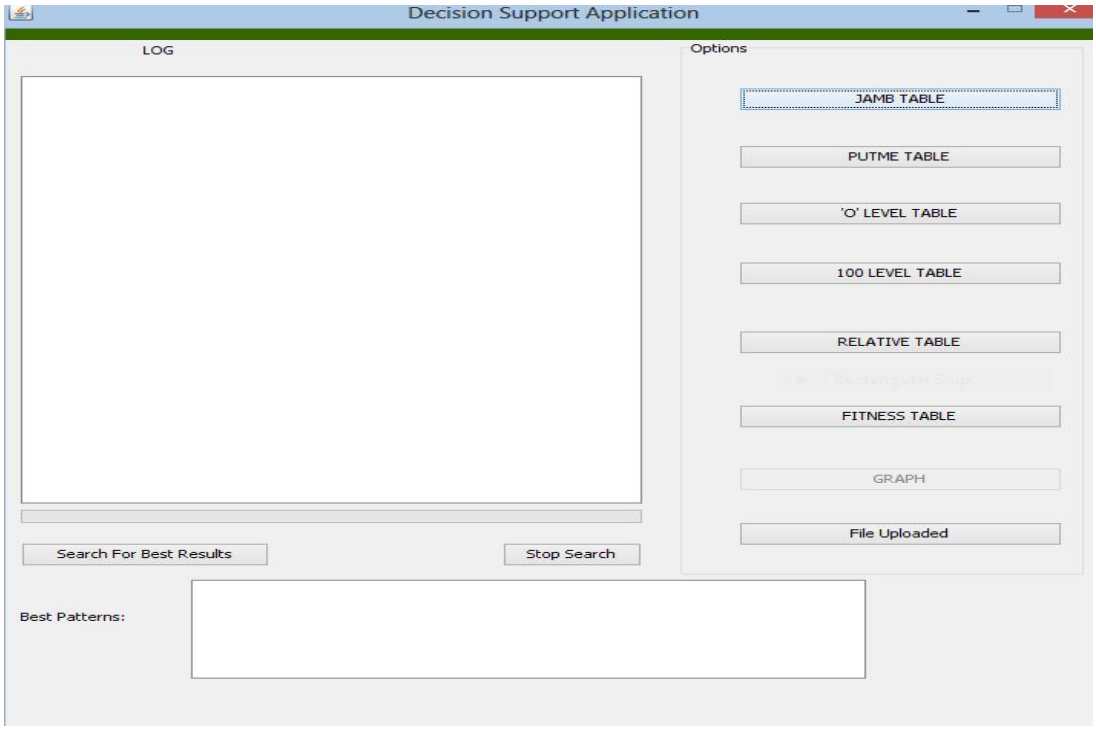


Figure 4.7: System interface for searching

The procedure that handles the section of searching, using the genetic algorithm approach, is given in the pseudo code 1:

Get number of selection
Initialize parent_population_list
Initialize child_population_list
Define max number_of_ iteration
Counter = 0;
Selected List: (best based on number of selection and the fitness table)
Parent_population_list← selected list

```
While (parent_population_list != child_population_list && counter<=number_of_
iteration)
{
        Perform GA operation on the Parent_population_list
                Crossover_list = Perform crossover on the Parent_population_list
                Mutation_list = Perform mutation on the Parent_population_list
        SET list = Parent_population_list + crossover_list + mutation_list
        New_SET_list = get the fitness rank of each SET list value
        If (New_SET_list value is found in Fitness Table)
        {
        Child_population_list = value
        }
        Parent_population_list = Child_population_list
        Increment in counter
}
Print the Child_population_list
```

Pseudocode 1: Genetic Algorithm step by step instructions

4.5 Result and Discussion

In this section, the result of the application is displayed and analyzed. The data mining technique used was a simple searching and classification techniques called GA. It is a well-trained predictive, classifier model that can be used to search and analyze the best pattern of students in their entry qualification based on the 100 level CGPA. The data set used in this thesis was obtained from records of Computer Science students kept in the department of mathematics and MIS unit of Ahmadu Bello University, Zaria, Kaduna State. The data sets of 651 students' records were collected for 2009/2010, 2010/2011 and 2011/2012 academic sessions in the department. Here two data sets were used. The first are the students UTME, PUTME and O'level used in gaining admission into the university. The second was the raw CGPA scores after the first year exams.

4.5.1 Configuration of the experiments

All results reported for GA were obtained with the same parameters. Though there is no straightforward way to configure the parameters of a genetic algorithm, we obtained good results with values:

1. Population size**:** 2 × number of each problem
2. Initial population: classifiers calculated by the fitness rule
3. Top (best): up to 1% from the previous population classifiers are copied to the next generation
4. Mutation rate: 0.05
5. Crossover probability: 0.7
6. Termination criterion: Maximum number of iteration of 1000

The result of searching for the best pattern is shows in figure 4.8.

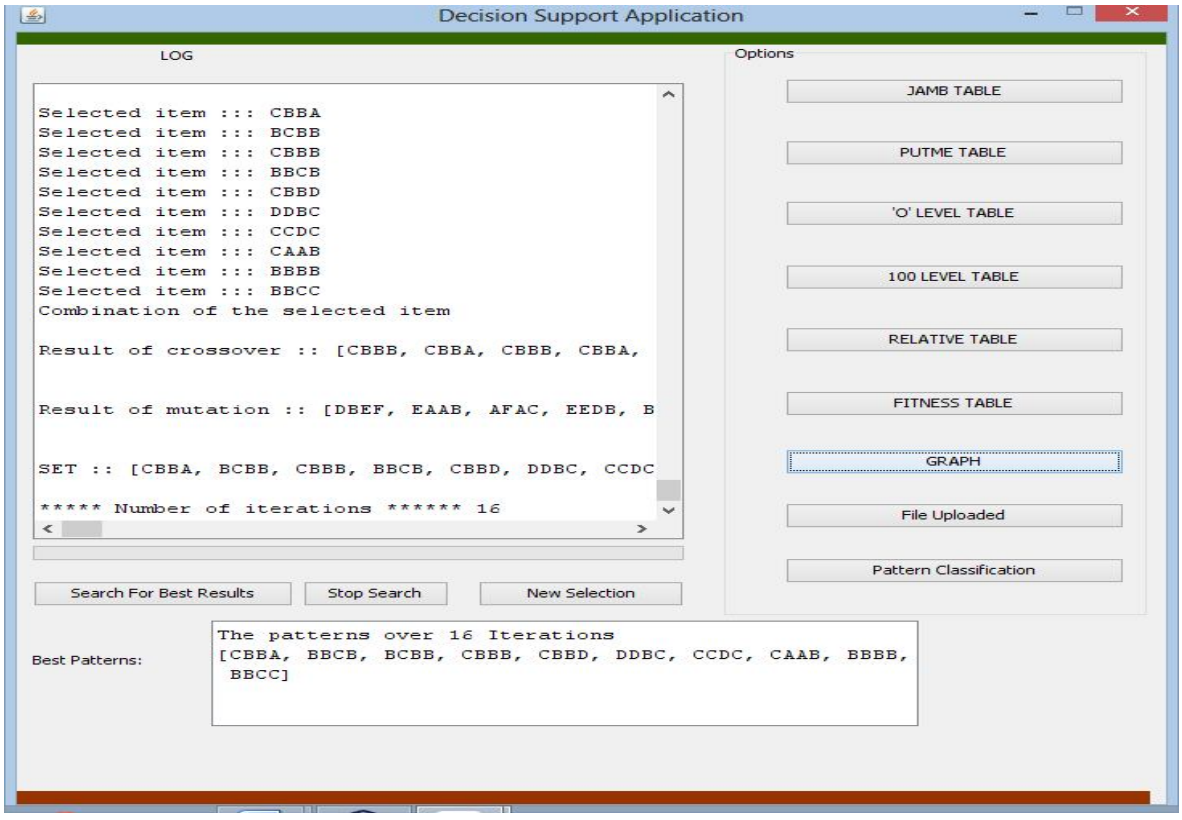


Figure 4.8 : Output result of best pattern

During the course of the searching, at sixteen iterations the best patterns were outputted as shown in Figure 4.8. The average of the best patterns were calculated and displayed in a graph as shown in the Figure 4.9 and table 4.1, respectively.

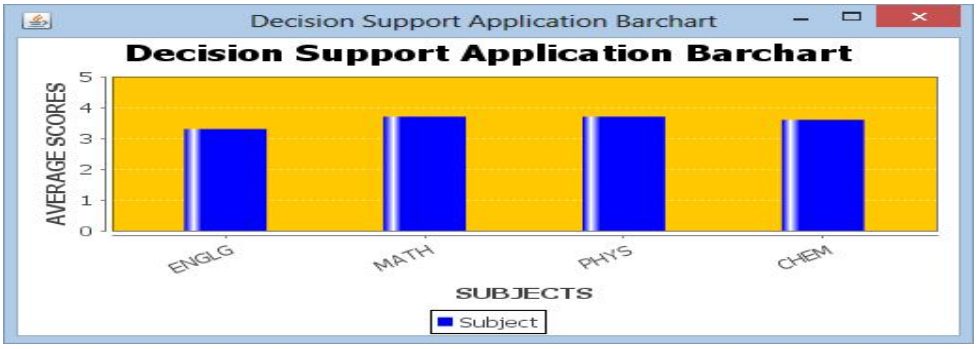


Figure 4.9: Graphical display of average score for each subject

Table 4.1: Table of average score performance

Subjects	**Average Scores**
English	3.3
Mathematics	3.7
Physics	3.7
Chemistry	3.6

Evidences of pattern from this study after several iterations revealed that credits in all subjects are important for good performance in 100 level. And since mathematics and physics have a good occurrence, as the students with best combinations of the results tends to have a very high performance in both subject combinations compare to other subjects in all selections. Based on the findings of the study, it was recommended that more emphasis should be placed on the good performance of Mathematics and physics results from the combined examinations as entry qualifications in admitting students into the bachelor of Computer science degree in the Department of Mathematics. The work has shown that, physics and Mathematics should continue to be a compulsory

requirement for admission in to the Department. This is in partial agreement with most literature that conclude that Mathematics and Physics has a significant relationship with students' performance in their degree examinations.

4.6. Validation

The main aim of this thesis is to establish a pattern using Genetic Algorithm for effective decision making in admission process in the University, using Computer science of A.B.U, Zaria as a case study. Correlation and average performance in the entry results was determined, in order to help in determining the appropriate weight for our alpha and to show the degree of association for each examination. The mean was determined to validate our assertion on the patterns average.

4.6.1 Correlation

The association of students' performance in O'level, UTME and PUTME was determined against the CGPA at the end of their 100level. Table 4.2 shows the corresponding relationship.

Table 4.2: PUTME, UTME and SSCE correlation relationship

PUTME	**Average**	**CGPA**
Average	1	
CGPA	0.203391	1

SSCE	**Average**	**CGPA**
Average	1	
CGPA	0.700703	1

UTME	Average	CGPA
Average	1	
CGPA	0.338946	1

It was found that SSCE has higher degree of association.

4.6.2 Mean

Statistical analysis was carried out to validate the result and the conclusion of the patterns. The good predictors of student performance were established to be physics and mathematics having the prominent features among the best pattern for several iterations Descriptive statistic was used to validate the data using SPSS application. The statistical result is display in table 4.3 and figure 4.10.

.

Table 4.3: Result from statistical value using descriptive Statistics

	N	Mean		Std. Deviation
	Statistic	Statistic	Std. Error	Statistic
ENG	138	2.50	.070	.827
MATH	138	3.33	.090	1.054
PHY	138	3.26	.086	1.010
CHEM	138	2.90	.098	1.148
Valid N (listwise)	138			

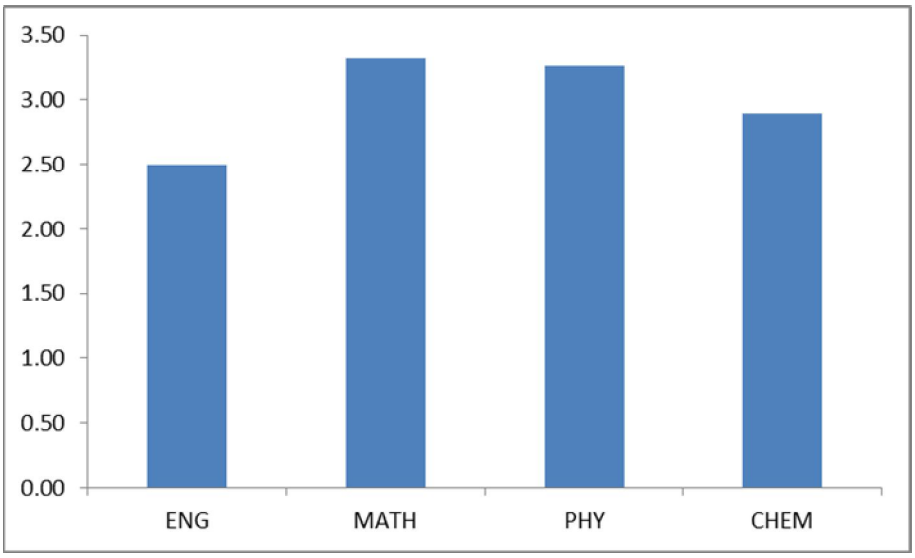


Figure 4.10: Graphical result of statistical analysis

The result of the GA was validated with the statistical analysis. The result shows that GA pattern where physics and mathematics are having prominent features is valid for decision making in admitting students.

4.7 Conclusion

The result obtained from the analysis of students' academic performance has showed that the Genetic algorithm application was able to model the data set used. It can be concluded from the result of the analysis that the current attention that is given to mathematics and physics as one of the basic requirement into the department as a variable in the selection process is a step in the right direction. It is expected that this work will serve as the basis of a more detailed and elaborate study involving more data from different Departments and by extension to other Universities.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 Summary

Educational data mining is used to study the data available in the educational field to bring out the hidden knowledge from it. Over the years, several techniques have been used to analyze students' performance from different points of view. This thesis presents data mining in education environment that identifies students' best patterns for admission into the University using Genetic Algorithm technique. The evaluation was done using undergraduate students' admission results into computer science, department of Mathematics, Ahmadu Bello University. The model was developed based on some selected input variables from the pre-admission data of three different session of students' offered admission into computer programme. Three examinations (UTME, PUTME and O'level) were used to group the students. The combination of these multiple examinations leads to a significant accuracy to support the decision. This thesis established a performance patterns for the first year entry results of students' in the University. The findings reveal that Mathematics and Physics have prominent features among the best pattern for almost all the selection.

5.2 Conclusion

This thesis used searching technique that identifies students' best patterns for admission into the University using Genetic Algorithm technique. The result obtained from the analysis of students' academic performance corroborate most of the literature assertions, thus, it showed that the Genetic algorithm application was able to model the data set used. It can be concluded from the result of the analysis that the current attention that is given to mathematics and physics as one of the basic requirement into the department as a variable in the selection process is a step in the right direction.

5.3 Recommendations

This thesis shows that Genetic Algorithm has the capability to bring out hidden knowledge from admission system. Hence, the searching can be giving more strength if all the factors that constitute to student academic performance are provided.

The limitation of this model stems from the fact that not all the relevant factors that play key roles in student performance were available to the researcher. The research was based on academic achievement only, factors like student background, age, number of years spent before getting admission, motivation, financial aids, and so on were not at the researcher disposal. Adding some of these factors will no doubt have influence on the pattern.

An extension of this research should be made to other departments. The current admissions system should be reviewed in order to improve the standard of candidates being admitted into the institution. Genetic algorithm with other machine learning tool should be combine which will be very useful for such an exercise.

REFERENCES

Adeniyi, O.S., Araoye, M.A., Amali, E. O., Eru, E.U., Ojabo, C. O., and Alao, O.O. (2010). Effect of Using Combination of O'level Result with JAMB Score on Student Performance in the First Two Years of Medical School in Benue State University, Makurdi. Afr. J. Biomed. 13(3), 189 – 195. Retrieved from http://www.bioline.org.br/request?md10031. Accessed on 30/11/2012.

Adewale, O., Adebiyi, A., and Solanke, O. (2007). Web-Based Neural Network Model for University Undergraduate Admission Selection and Placement.Pacific Journal of Science and Technology, 8(2), 367-384. Retrieved from http://www.akamaiuniversity.us/PJST8_2_367.pdf. Accessed on

Adeyemi, T. O. (2010). Credit in Mathematics in Senior Secondary Certificate Examinations as a Predictor of Success in Educational Management in Universities in Ekiti and Ondo States, Nigeria. Middle-East Journal of Scientific Research, 5 (4), 235-244. Retrieved from http://www.idosi.org/mejsr/mejsr5%284%29/8.pdf. Accessed on 22/8/2013.

Adeyemo, A. B., and Kuye, G. (2006). Mining Students' Academic Performance using Decision Tree Algorithms. Journal of Information Technology Impact, 6(3), 161-170. Retrieved from http://www.jiti.com/v06/jiti.v6n3.161-170.pdf

Aigbe, P.E. (2008). Design and Implementation of a Web Based University Admission and Placement Neural Network Model. Unpublished M.Sc. Thesis. Ahmadu Bello University Zaria, Kaduna State.

Ajala, O. P. (2010). Three Years of Post UME Screening: Influence on Science Education Students' Achievement. Delta State University, Abraka. International Journal Education Science, 2(1), 29-40. Retrieved from

http://www.krepublishers.com/02-Journals/IJES/IJES-02-0-000-10-Web/IJES 02-1000-10-Abst-PDF/IJES-02-1-29-10-031-Patrick-A-O/IJES-02-1-29-10-031 Patrick-A-O-Tt.pdf

Akinola, O.S., Akinkunmi, B.O., Alo T.S. (2012). A Data Mining Model for Predicting Computer Programmeming Proficiency of Computer Science Undergraduate Students. Afr J Comp and ICT, 5(1), 44-52. Retrieved from http://www.ajocict.net/uploads/V5N1P3__2012_AJOCICT__A_Data_Mining_ Model_for_Predicting_Computer_Programming_Proficiency.pdf. Accessed on 5/10/2012.

Ande, E. (2006). Obasanjo accuses JAMB of corrupt practices. The Guardian, June, 13, P 7. Retrieved from http://www.krepublishers.com/02-Journals/JSS/JSS-22 0-000-10-Web/JSS-22-2-000-10-Abst-PDF/JSS-22-2-101-10-951-Ifedili-C-J A/JSS-22-2-101-10-951-Ifedili-C-J-A-Tt.pdf

Antonio, B. (2013). A comparative study of crossover operators for Genetic Algorithms to solve the job shop scheduling problem. Wseas transactions on computers 4, 164 - 173

Bandyopadhyay, S., and Saha, S. (2013). Some single and multiobjective optimization techniques. Springer-Verlag, DOI 10.1007/978-3-642-32451-2_2

Barricelli, N. A. (1957). Symbiogenetic evolution processes realized by artificial methods. Methodos, 9(35-36), 143-182. Retrieved from https://svn-d1.mpi inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf

Barricelli, N. A. (1962). Numerical testing of evolution theories. Acta Biotheoretica, 16(1-2), 69-98. Retrieved from https://svn-d1.mpi inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf

Bean, J. P. (2005). Nine themes of college student retention. College student retention:

Formula for student success, 215-243.

Behrouz, M.B., and William, F. P. (2003). Using Genetic Algorithms for Data Mining Optimization in an Educational Web-Based. System. Springer-Verlag Berlin Heidelberg . 27(24), 2252–2263. Retrieved from http://www.lon capa.org/papers/v90-gapaper.pdf. accessed on 24/5/2014.

Binitha, S., and Sathya, S. S. (2012). A Survey of Bio inspired Optimization Algorithms. International Journal of Soft Computing and Engineering, 2(2) 2231-2307. Retrieved from http://www.ijsce.org/attachments/File/v2i2/B0523032212.pdf

Bledsoe, W. W. (1961). The use of biological concepts in the analytical study of systems. In Proceedings of the ORSA-TIMS National Meeting. Retrieved from https://svn-d1.mpi inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf

Boufardea, E., and Garofalakis, J. (2012). A Predictive System for Distance Learning Based on Ontologies and Data Mining. The Fourth International Conference on Advanced Cognitive Technologies and Applications 151-158. Retrieved from https://www.google.com/url?q=http://www.thinkmind.org/download.php%3Farticleid%3Dcognitive_2012_7_30_40125&sa=U&ei=y-NwU6AygZvIBPj8gbgH&ved=0CAgQFjAC&client=internal-uds-cse&usg=AFQjCNGbmEbG0fsGWjC7bS8hp3TXWSnC0g

Box, G. E. (1957). Evolutionary operation: A method for increasing industrial productivity. Applied Statistics, 81-101. Retrieved from https://svn-d1.mpi inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf

Bremermann, H. J. (1962). Optimization through evolution and recombination. Self organizing systems, 93-106. Retrieved from https://svn-d1.mpi

inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf

Department of Mathematics, Ahmadu Bello Zaria. Second semester 2009/2010, 2010/2011 and 2011/2012 result summary.

Dimitris, K. (2005). Analyzing student performance in distance learning with genetic algorithms and decision trees. 1-30. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.7658&rep=rep1&type=pdf

Ebiri K. (2006). Post JAMB basis for admission says Obasanjo. Guardian, April 13,P.49. Retrieved from http://www.krepublishers.com/02-Journals/JSS/JSS-22-0-000-10-Web/JSS-22-2-000-10-Abst-PDF/JSS-22-2-101-10-951-Ifedili-C-JA/JSS-22-2-101-10-951-Ifedili-C-J-A-Tt.pdf

Elden, A., Moustafa, M. A., Harb, H. M., and Emara, A. (2013). Adaboost Ensemble with Simple Genetic Algorithm for Student Prediction Model. International Journal of Computer Science, 5(2), 73-85. Retrieved from http://airccse.org/journal/jcsit/5213ijcsit07.pdf. Accessed on 24/10/2013.

El-Emary, I. M., and El-Kareem, M. M. A. (2008). On the application of genetic algorithms in finger prints registration. World Applied Sciences Journal, 5(3), 276-281. Retrieved from http://www.idosi.org/wasj/wasj5%283%29/3.pdf. Accessed on 30/7/2012.

FGN (2004). National policy on education, NERDC Yaba, Lagos 7, 44-45.

Fogel, L. J., Owens, A. J., and Walsh, M. J. (1966). Artificial intelligence through simulated evolution. Chichester: John Wiley and Sons, 1-170

Fraser, A. S. 1957. Simulation of genetic systems by automatic digital computers: Effects of linkage on rates of advance under selection. Australian Journal of Biological Science 10, 492–499.

Friedman, G. J. (1959). Digital simulation of an evolutionary process. General Systems Year book, 4, 171-184.

Goldberg, D. E. (1989). Genetic algorithms and Walsh functions: A gentle introduction. Complex Systems 3, 129–152.

Graham, L. D. (1991). Predicting academic success of students in a master of business administration programme. Educational and Psychological Measurement, 51(3), 721-727.

Harb, H. M., and Moustafa, M. A. (2012). Selecting Optimal Subset of Features for Student Performance Model. International Journal of Computer Science, 9(5). Retrieved from http://ijcsi.org/papers/IJCSI-9-5-1-253-262.pdf

Holland, J. H. (1973). Echoing emergence: Objectives, rough definitions, and speculations for echo class models. In G. Cowan, D. Pines, and D. Melzner, eds., Complexity.Addison−Wesley.

Holland, J. H. (1975). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Michigan Press.

Jadrić, M., Garača, Ž., and Čukušić, M. (2010). Student dropout analysis with application of data mining methods. Management: Journal of Contemporary Management 15(1), 31-46. Retrieved from http://www.efst.hr/management/Vol15No1-2010/2-Jadric_Garaca_Cukusic-final.pdf

Jusoff, K., Samah, S. A. A., and Isa, P. M. (2008). Promoting university community's creative citizenry. Proceeding of world academy of science, 2(7), 1-6. Retrieved from http://waset.org/publications/3024/promoting-university-community-s-creative-citizenry

Kaur, K., Chhabra, A., and Singh, G. (2010). Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system. International journal of computer science and security, 4(2), 183-198. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.177.665&rep=rep1&type=pdf

Keshavamurthy, B. N., Khan, A. M., and Toshniwal, D. (2010). Improved genetic algorithm based classification. International Journal of Computer Science and Informatics 1(3), 2231 –529. Retrieved from http://interscience.in/IJCSI_Vol1Iss3/1.pdf. Accessed on 24/5/2012.

Leyva, L. J.C. (2005). A Genetic Algorithm Application for the Individual and Group Multicriteria Decision Making: PhD Thesis Resume. Computacióny Sistemas, 4, 183-188. Retrived from http://www.scielo.br/pdf/pope/v25n1/24250.pdf. Accessed on 28/1/2013.

Lievens, F., and Coetsier, P. (2002). Situational tests in student selection: An examination of predictive validity, adverse impact, and construct validity. International Journal of Selection and Assessment, 10(4), 245-257.

Luan, J. (2004). Data Mining Applications in Higher Education. New directions for Institutional Research, 113. https://courses.cs.washington.edu/courses/cse590d/04wi/dataminighigh.pdf

Lubna, Z.B. (2009). University Admission System using Machine Learning. Eng.and Tech. journal, .27(16), 1-14. Retrieved from http://www.uotechnology.edu.iq/tec_magaz/volume272009/No.16,2009/researches/Text%20%2812%29.pdf. Accessed on 24/10/2013.

Magnus, E. H. (2003). Genetic Algorithms for Rule Discovery in Data Mining. Retrieved from

http://www.hvasslabs.org/people/magnus/schoolwork/mining/project1/ga-rule.pdf. Accessed on 7/10/2012

Marchima, M. R. Noor, H. Ibrahim, T. Nor, S. M. and Shahriman, N.(2011). Fault Prediction Model for Web Application Using Genetic Algorithm. International Conference on Computer and Software Modeling, 1-7. Retrieved from http://www.ipcsit.com/vol14/14-ICCSM2011-S0042.pdf. Accessed on 30/7/2012.

Martin, F. G., and Cockerham, C. C. (1960). High speed selection studies. Kempthorne, ed., Biometrical Genetics. Pergamon.

Melanie M. (1999). An Introduction to Genetic Algorithms. A Bradford Book the MIT Press, London. Fifth edition.

Naik, B., and Ragothaman, S. (2004). Using Neural Networks to Predict MBA Student Success. College Student Journal, 38(1), 143.

NUC (2004). Report on the performance of the Federal University system in 2002 presented at the special meeting convened by the Honourable Minister of Education on Thursday, December 12.

Rechenberg, I. (1973). Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipiender Biologischen Evolution. Frommann−Holzboog (Stuttgart).

Reed, J., Toombs, R., and Barricelli, N. A. 1967. Simulation of biological evolution and machine learning. Journal of Theoretical Biology 17: 319–342.

Renner R.S. and Lacher R.C. (2000). Combining Constructive Neural Networks for Ensemble Classification. Proceedings of the Fifth Joint Conference on Information Science, 887-891

Salim B A. (2006). A case study for common and unified entrance examination into

Nigerian Universities. A paper presented at the 6th NAPEUAP Leadership Forum in Calabar.

Salahdeen H. M., and Murtala B. A. (2005). Relationship between Admission Grades and Performances of Students in the First Professional Examination in a New Medical School. African Journal of Biomedical Research, Ibadan Biomedical Communications Group. Retrieved from http://www.bioline.org.br/md. Accessed on 22/8/2013.

Sanjay, S., Pradeep, S., Manikanta, V., Kumara, S.S., and Harsha P. (2011). Genetic algorithm based approach for the selection of projects in public R&D institutions. Indian journal of computer science and engineering, 2(4), 523-531.

Sapna S.,Tamilarasi A. and Pravin M.K.(2012). Implementation of Genetic Algorithm in Predicting Diabetes. IJCSI International Journal of Computer Science, 9(3).

Schönbrunn K. and Hilbert A. (2006). Data Mining in Higher Education, Studies in Classification,Data Analysis, and Knowledge Organization Advances in Data Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V., Berlin.

Schwefel, H. P. (1975). Evolutionsstrategie und numerische Optimierung. Ph.D. thesis, Technische Universität Berlin.

Schwefel, H. P.(1977). Numerische Optimierung von Computer−Modellen mittels der Evolutions strategies. Basel: Birkhäuser

Seng, O., Bauyer, M., Biehl, M., and Pache, G.(2005). Search-based improvement of subsystem Decomposition. Proceedings of the Genetic and Evolutionary Computation Conference: 1045 – 1051.

Sheel, S.J, Renner R.S. and Dawsey, S. K. (2002). Alternatives to Math Placement

Exams: A Look at Discriminant Analysis, Neural Networks and Ensembles of Networks, Department of Computer Science, Coastal Carolina University, Chico Conway, SC USA.

Takale, A. (2004). Constructing Predictive Models to Assess the Importance of Variables in Epidemiological Data Using A Genetic Algorithm System employing Decision Trees (Doctoral dissertation, University of Minnesota).

Usama, F. and Ramasamy, U. (2002). Evolving Data Mining into Solutions for Insights, Communications of the ACM, 45(8), 28 – 31.

Wafa, S.A.(2013). Steady State Genetic Algorithm in University Admission Decision. BT technol Journal, 4(14), 32–36

Wilson, T. (1999). A Student Selection Method and Predictors of Success in a Graduate Nursing Programme. Journal of Nursing Education, 38, 183-187.

Wright, R.E. and Palmer, J.C. (1994). GMAT Scores and Undergraduate GPAs as Predictors of Performance in Graduate Business Programmes. Journal of Education for Business, 70, 344-348

APPENDIX

SOURCE CODE LISTING

```
public class DecisionSupportSystem extends javax.swing.JFrame {

    /** Creates new form DecisionSupportSystem */
    CardLayout card = new CardLayout();
 private int numberofselection;
private String[] subjects;
   public DecisionSupportSystem(String[] subj) {
       subjects = subj;
       initComponents();
        setTitle("Decision Support Application");
        ImageIcon img=new ImageIcon("GA.png");
        setIconImage(img.getImage());
        setLocationRelativeTo(null);
    }
   public String[] getSubjects()
   {
      return subjects;
   }
   public void setNumberofselection(int s)
   {
      numberofselection = s;
   }
   public int getNumberofselection()
   {
      return numberofselection;
   }
   public int getPopulationSize()
   {
      int numofrow = 0;
```

```
      try{
            Class.forName("com.mysql.jdbc.Driver");
            Connection c =
DriverManager.getConnection("jdbc:mysql://localhost/predictor", "root", "root");
            Statement s = c.createStatement();
            ResultSet r = s.executeQuery("select count(*) from jamb");
            r.next();
               numofrow = Integer.parseInt(r.getString(1));
               r.close();
               s.close();
               c.close();
          }catch(Exception ex){ex.printStackTrace();}
      return numofrow;
   }
    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel6 = new javax.swing.JLabel();
        jPanel4 = new javax.swing.JPanel();
        jPanel5 = new javax.swing.JPanel();
        jPanel3 = new javax.swing.JPanel();
        jLabel27 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jTextField1 = new javax.swing.JTextField();
        jButton3 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setResizable(false);

        jPanel1.setLayout(card);/*
        jPanel1.setLayout(new java.awt.CardLayout());
        */

        jLabel6.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/dsss/s.PNG"))); // NOI18N

        jPanel4.setBackground(new java.awt.Color(0, 102, 0));

        javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
```

```
        jPanel4Layout.setHorizontalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 712, Short.MAX_VALUE)
        );
        jPanel4Layout.setVerticalGroup(
            jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 15, Short.MAX_VALUE)
        );

        jPanel5.setBackground(new java.awt.Color(102, 51, 0));

        javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
        jPanel5.setLayout(jPanel5Layout);
        jPanel5Layout.setHorizontalGroup(
            jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 692, Short.MAX_VALUE)
        );
        jPanel5Layout.setVerticalGroup(
            jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGap(0, 19, Short.MAX_VALUE)
        );

        jLabel4.setFont(new java.awt.Font("Consolas", 1, 18));
        jLabel4.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        jLabel4.setText("SELECTION");

        jLabel5.setText("Number of Selection(The Best)");

        jButton3.setText("Start Genetic Algorithm");
        jButton3.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton3ActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
        jPanel3.setLayout(jPanel3Layout);
        jPanel3Layout.setHorizontalGroup(
            jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
```

```
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.T
RAILING, false)
                        .addComponent(jTextField1,
javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(jLabel5,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                    .addGap(127, 127, 127))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()
                    .addComponent(jLabel4,
javax.swing.GroupLayout.PREFERRED_SIZE,                                    204,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(110, 110, 110))))
            .addGroup(jPanel3Layout.createSequentialGroup()
              .addGap(26, 26, 26)
              .addComponent(jButton3,  javax.swing.GroupLayout.DEFAULT_SIZE,  204,
Short.MAX_VALUE)
              .addGap(94, 94, 94))

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
              .addGroup(jPanel3Layout.createSequentialGroup()
                .addGap(0, 74, Short.MAX_VALUE)
                .addComponent(jLabel27,   javax.swing.GroupLayout.PREFERRED_SIZE,
177, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(0, 73, Short.MAX_VALUE)))
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
              .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
              .addComponent(jLabel4,  javax.swing.GroupLayout.PREFERRED_SIZE,  40,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGap(55, 55, 55)
              .addComponent(jLabel5)
              .addGap(33, 33, 33)
              .addComponent(jTextField1,  javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addGap(46, 46, 46)
```

```
            .addComponent(jButton3)
            .addContainerGap())

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)
            .addGroup(jPanel3Layout.createSequentialGroup()
               .addGap(0, 132, Short.MAX_VALUE)
               .addComponent(jLabel27)
               .addGap(0, 121, Short.MAX_VALUE)))
      );

      jLabel1.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/dsss/GA.png"))); // NOI18N

      javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
      getContentPane().setLayout(layout);
      layout.setHorizontalGroup(
         layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
               .addGroup(layout.createSequentialGroup()
                  .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
480, Short.MAX_VALUE)
                  .addGap(27, 27, 27))
               .addGroup(layout.createSequentialGroup()
                  .addGap(178, 178, 178)
                  .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
249, javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))
            .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE,
205, javax.swing.GroupLayout.PREFERRED_SIZE))
         .addComponent(jPanel4, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
         .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addContainerGap())
         .addGroup(layout.createSequentialGroup()
            .addGap(43, 43, 43)
            .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 669,
Short.MAX_VALUE))
      );
```

```
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1,        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jPanel4,        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
                    .addComponent(jLabel6,     javax.swing.GroupLayout.PREFERRED_SIZE,
220, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jPanel3,     javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jLabel1,        javax.swing.GroupLayout.PREFERRED_SIZE,
250, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jPanel5,        javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        );

        pack();
    }// </editor-fold>



private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jTextField1.getText().matches("[0-9]+")  )
    {
      if(Integer.parseInt(jTextField1.getText())            >            1                        &&
Integer.parseInt(jTextField1.getText()) < getPopulationSize() )
       {
        setNumberofselection(Integer.parseInt(jTextField1.getText()));
        System.out.println("The number of selection is "+getNumberofselection());
        this.dispose();
        System.out.println("The number of selection is "+getNumberofselection());
        new Results( getNumberofselection(), getSubjects()).setVisible(true);

        }else{
```

```
JOptionPane.showMessageDialog(this, "Selection number must be between 1 and
population size");
      }
    }
    else{
       JOptionPane.showMessageDialog(this,"Invalid Selection number",null,
JOptionPane.ERROR_MESSAGE);
    }

}

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

            }
        });
    }
```