Solving the Problems of Timetable Using Genetic Algorithm Case Study:

Faculty of Information Technology Timetable

By

Eslam A. Alaraibi

Supervision

Dr. Omar M. Sallabi

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of computer Science

University of Benghazi

Faculty of Information Technology

Fall 2017

University of Benghazi
Faculty of Information Technology
Department of Computer Science

Title of Thesis

Solving the Problems of Timetable Using Genetic Algorithm
Case Study: Faculty of Information Technology Timetable

By
Eslam A. Alaraibi

Approved by:

Dr. Omar Mustafa Elsalabi ....
Advisor
Graduate Examination Committee

Dr. Abdelhamead M. Abdelkafy ....
External examiner

Dr. Dr. Ahmed O. Lawgali ....
Internal examiner

Countersigned by:

Khirallah S. Alferjani	Tawfig M. Eltaweel
Head, Computer Science Department	Dean, Faculty of Information Technology

Acknowledgements

First of all, I would like to thank Almighty Allah for blessing me with his mercy, and the strength to successfully carry out this work. I would like to thank and appreciate my mother, father and brothers. My family provided a stable environment allowing me to accomplish this work and carry with me the hardships and difficulties of study. I was very helpful in every step of the study to completion of the graduate studies. Throughout my thesis, I received invaluable guidance, encouragement and fantastic support from my thesis advisor, Dr Omar M. Sallabi. He presented His guidance, help and suggestion made me more confident to overcome many difficulties during my research so I can complete this thesis on time. I would like to thank him and I am very great full of gratitude to him. It is my pleasure to acknowledge the University of Benghazi, Benghazi - Libya, for their sponsorship to my Master's study. I would also like to special thanks to all the members of the Faculty of Information Technology and Computer Science Department at the University of Benghazi.

In conclusion, I would like to express my thanks to everybody who contributed to this accomplishment in a direct or indirect way. I thank them all who made this thesis possible.

Dedicated to

The spirit of my father,

My beloved mother,

My family,

My friends...

Contents

List of Tables

List of Figures

Abstract

The timetabling problem is one of the real world's optimizations problems, and classified as a Non-deterministic Polynomial (NP) and referred as “NP-hard” problems. In this thesis we used genetic algorithms, a technique of artificial intelligence, to find solutions to the timetable problem, thus obtaining timetables without conflicts and taking into account the conditions of improvement in the tables, which make the student and teacher table in the best possible situation. Nevertheless, get multiple solutions (tables, you can choose the best table that meets the requirements best).

The proposed tool is used to automate the academic timetable for the faculty of information technology, Benghazi University, which is can be used to optimize and utilize the faculty resources, for both lecturers and teaching venues. In terms of optimization, we find encouraging results obtained from the tool compared with relevant results from other researchers, the results show the ability of the tool to generate more tables with less time.

Chapter 1

Introduction

1.1 Overview

The rapid development of the contemporary world has led to the emergence of complex problems that need to be tackled using high reliable techniques. Recently, the interest in the science of Artificial Intelligence (AI) has increased because of its worthiness and effectiveness in handling and resolving many complex problems, especially those with a large number of variables and restrictions. Solving such problems with traditional methods may not lead to sufficient solutions. The field of Artificial intelligence (AI) specializes in solving complex problems that are classified as a NP-hard problem (S. Even et al. 1976), where it is difficult to solve this kind of problems by using the traditional software. It is also interested in finding the best solution for different and complex problems.

The goal of this thesis is to examine several approaches to (AI) algorithms and develop a tool to solve the problems of courses and exams' timetable.

According to Qu et al. (2009), there are several approaches that can be followed to resolve these problems, such as; Heuristic approach, Simulated annealing, Graph coloring approach, Network streaming approach, knowledge principles approach, Mathematical, Logical constraints approach, Tuba search, and genetic algorithms approaches.

The problem of planning an academic schedule is one of the problems that can be found in many solutions where they are considered a wide range of optimization problems (Ms.Jayashree et al. 2014) and should work to find out the best solution of the proposed solutions. Many universities have problems with the process of preparing academic timetables (Can Berk et al. 2011and R. Lewis 2006), where people have to arrange them manually, thus; this consumes a lot of time. Universities used a normal programming (non-smart) to solve the problem of preparing timetables, but the problems have not been solved and the result of schedules was not perfect. This has created the challenge of automating this problem(R. Lewis 2006). Moreover, many problems remained unresolved (Not better than what it could have been). Therefore, this research is using the "Genetic Algorithms" which is one of Artificial Intelligent (AI) techniques to find the solutions to timetabling problems (Sani et al. 2016; Sadaf Naseem 2012). Thus; we can get a timetable without any conflicts, as well as; improving the schedule conditions to make teachers and their table in the best possible attitude.

Genetic algorithms technique has been used in this research to get several solutions (the best timetable that can be chosen with the best requirements).

1.2 Problem Statement

The main problem of preparing schedules is being dependent on finding the best solutions from several available solutions due to the many possibilities that can be accessed.

The timetable is one of "NP-hard problem" problems because of its complexity in setting the lectures classes, the difficulty of flexible constraints and increases the size of the timetable which is more complicated (M. R. Garey et al. 1979). All of these problems because the work on the timetable is done manually by employees.

- The planning problem for timetables that contains a large number of lectures which require accessing the best timetable from the tables resulting. Moreover, we need to change this schedule from time to time.
- The scheduling work is consuming a lot of time and effort, timetable designers are facing a lot of problems in organizing them because of many possibilities in timetable constraints.
- Working to resolve the timetable problems (courses and exams) at the faculty of Information Technology at Benghazi University, by design and implement a system that works to find a radical solution to the problem.
- The planning of timetables requires highly skilled people and experienced in preparing timetables, sometimes we do not find them when we need to change the academic schedule.

1.3 Scope of Study

The scope of this research is limited to the following:

- Study the genetic algorithms and choose a reliable technique to improve the process of preparing timetables.
- Develop an application to prepare schedules that take into account the massive of restrictions, especially the faculty of information technology.
- Improve the process of subjects' allocation in the schedule.
- Help request for inquiries about the available classes times and lecturers.

1.4 The Aims

The thesis aims to determine the genetic algorithms approach by applying new operations and new technique and applying the selected approach to solve the timetable problem.

1.5 The Objectives

The thesis objectives are summarized as the following:

- To examine the approaches of genetic algorithms.
- Develop a general algorithm that can be used in solving the problem.
- Create timetables easily and less complicated without any conflicts by accepting several constraints and preferences.

1.6 Thesis organization

• **Chapter Two**: This chapter describes the problem with an explanation and clarification of the scheduling process, along with their restrictions on the methods which are used to solve the problem. In addition, this chapter explained genetic algorithms and graphics theory that will be used in schedules problem.

• **Chapter Three:** Presented previous researches and studies were done in solving the timetabling problem and the techniques of solution used to solve in the same type of problem.

• **Chapter Four:** This chapter provides a detailed analysis of the problem, and then design an algorithm using Genetic algorithm techniques. Finally, an application for solving the schedules of timetables will be developed using programming techniques.

• **Chapter Five:** This chapter provides a case study to demonstrate and illustrate the results, contains graphs and charts showing the various results of simulations for IT-Collage timetable problem. This chapter also shows the discussion and evaluation of the results.

• **Chapter Six**: This chapter presents a summary of this work, the results with final discussion, the application features. Also, this chapter will explain the thesis goals how are achieved, the contribution, the future work and limitations of our work.

Chapter 2

Background

2.1 Overview

In this chapter, we will present the academic timetable and its limitations during the preparation process, and the extensions of necessary requirements for universities. Also, we discussed the general principles of genetic algorithms and graphics theory which will be used in this research to solve the problem of timetabling.

2.2 Timetabling:

The problem that we have focused on is how to create a timetable i.e. (Lectures and exams' timetable). The problem of timetabling is known as one of the most common problems in the real world which faces most educational institutions, that requires high accuracy of distribution of subjects.

Khairul Anwar et al, (2013) defined "Timetabling as a process of assigning certain resources or events to the limited time slots and rooms according to a set of constraints." Generally, the timetable can be defined as the process of making a decision on how to allocate subjects between the diversity of available tasks.

The complexity of timetabling is in the preparation process, which entirely depends on the number of restrictions. However, in the case that there are limited resources and lack of time, the process of planning of timetabling will be extremely difficult.

Educational timetabling is a problem, which involves creating a schedule for student groups, rooms and teachers to satisfy certain constraints, educational timetabling includes examination timetabling, school timetabling and University courses timetabling (Pillay 2014). There are other timetabling problems such as transportation, healthcare and sports (Petrovish & Burke 2004).

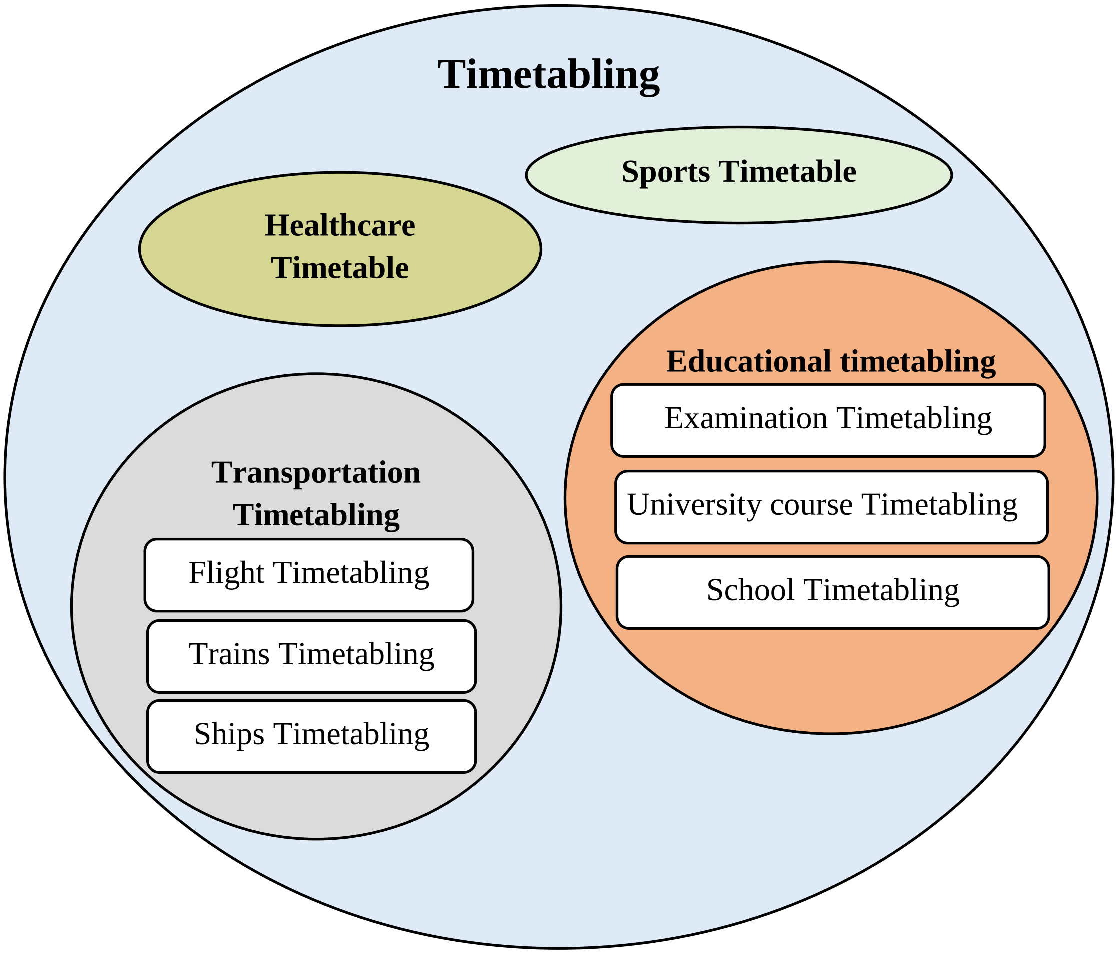


Figure 1: An overview of timetabling problems

Almost, the study and examination offices at colleges face complex problems, like conflicts in hours in the process of preparation of timetabling. The reason of the appearance of such problems are teachers' tables and their teaching engagements, the large number of students and the lack of classrooms and laboratories. Because of all these problems, timetables become sameness and constant and there is no way to change them to become better. That leads to negative affect on students, and sometimes the timetable becomes very stressful for them and other times it becomes cumbersome for teachers.

From our reality of our daily lives in our university and according to the problems that we have faced in the lectures times, which were very tiring and there was no possibility to change them because this process is considered as a difficult, and a complex task. It also takes a lot of time even if it is used the conventional programming.

The existing problems in the process of preparing tables are summed up in the following points:

- The process of preparing a timetable is very complex, the conventional programming gives solutions but these cannot be considered as optimum solutions because they do not put an end to all the problems in the timetable.
- The timetable is different from college to another and from university to another that is because of their different needs. For example, there are colleges operating in a year system and others operating in a semester system and also every college has different subjects.
- The process of planning a timetable is done manually and this is what makes it complicated and difficult. Moreover, it must be taken into account the restrictions of timetable.

From the above, it is obvious to us that the process of preparing tables consumes a lot of time and a lot of mental and physical effort. Especially, these tables must be created every semester or every year. In case that the timetable is for the semester system that means that these tables should be created twice in an academic year. This makes these timetabling constant and unalterable because they are done once in a year and remains for a long period of time, which does not correspond to the change in the number of the students that substantially increase. The stability of these tables causes to make them uncomfortable and unsuitable for teachers and students as well.

We can define the timetable as a table which elucidates lecture times for teachers and students by locating the lecture (classroom) and specify the time for every subject. The process of organising tables is for the whole semester.

Schearf (1999) classified educational timetabling into the following three main categories:

- School timetabling.
- Examination timetabling.
- University course timetabling.

They share the same basic characteristics of the general timetabling problem but can still have significant differences between them. Each one of them has its own constraints, requirements, and rules. More details on educational timetabling can be found in different survey papers, see R. Lewis (2008) and R. Lewis (2006).

2.2.1 School Timetabling:

It is a timetable which is made throughout the week, and takes into account the conflicts of lessons time for teachers, and assigns certain times every day for subjects that will be taught, and so to produce a timetable which satisfies all the constraints.

2.2.2 Examination Timetabling:

Syariza et al. (2014) Explained that the examination timetabling problem can be defined as:

"The assignment of a finite set of examinations to a finite set of time-slots and set of rooms, satisfying various constraints without a conflict" The examination timetabling problem refers to the assignment of time slots and rooms so that students can take examinations without clashes.

2.2.3 The University Courses Timetabling:

According to Carter et al (1998), the University Course Timetabling (UCTP) is defined as "A multidimensional assignment problem, in which students and teachers (or faculty members) are assigned to courses, course sections or classes and events (individual meetings between students and teachers) are assigned to classrooms and time slots".

UCTP is represented as sets of events (activities), time slots and rooms. The events are scheduled to time slots and rooms. A student group is a set of students taking the same courses. Student groups and teachers attend a subset of all events. UCTP is an optimisation problem to create a timetable for universities in the beginning of each semester to satisfy requirements of courses, rooms, students and teachers described by constraints. Algorithms to solve UCTP aim to minimise the amount of violated constraints but the definition of the problem and constraints often differs between universities. Where we must work on a timetable that satisfies the constraints to provide a conducive learning environment.

2.3 University timetabling problem:

2.3.1 Constraints Timetable:

We should take into account these essential restrictions when we work on such tables:

- The duality of subjects is not allowed in the same place and time; in other words, it cannot be more than one lecture at the same time in the same room.
- Have to know how many hours does each subject have and when these subjects are put into the timetable should delimit the lectures distribution in the week.
- The regular student should not be taught more than four or five hours in a day.
- Should associate the teacher with his/her subject in the timetable to avoid occurring conflicts for the teacher in teaching hours where he/she cannot teach two subjects at the same time.
- Should associate classrooms with subjects to avoid the duality with another subject in the same room and at the same time, so it should be a special table for each room to clarify when it is available.
- Should allocate a classroom for each lecture, so the room becomes associated with the lecture.
- It should not be a long break of time between the times of lectures also they should not be connected, but there should be a short interval between them.
- Classrooms and laboratories should be at the same building or close to each other so as not to waste students' time on moving between them.
- The day in summer is long whereas it is short in winter, so the timing of lectures should base on seasons.
- Regarding the examination timetabling, that must be taken into account the conflicts at the exam time for students and teachers. To put more than one exam for the student is not allowed as well as the teacher.

- Should distribute the exams' time during the period of the examination timetable. As not to produce an exam timetable which is exhausting to student and teacher by too many exams in consecutive days or more than one exam on the same day.

2.3.2 Timetable requirements:

There are many things need to be determined before starting to prepare timetables:

- Have to know and decide how many subjects that will be taught not only in the semester but in the whole academic year. Additionally, specify whether these subjects need classrooms or laboratories in order to avoid conflicts.

- We have to specify how many hours for each subject and divide them into two periods or more in the week.

- Should specify how much the classroom can carry and how many students we have got. By standards of university studying quality that should not be in the room more than 35 students. However, this standard cannot be achieved because of a large number of accepted students and the lack of university's resources.

- If the number of students was large, we have to divide them into groups of the same subject in different times, and we have to stop dividing them according to classroom bearing.

- Should specify how many subjects will be taken in the final examinations.

- Should specify how many students will enter the exams and the subjects.

2.4 Genetic Algorithm

Genetic Algorithms (GAs) One of the techniques used in evolutionary computation (EC) research, which basic principles of genetic algorithms were promoted by Holland, J.(1975). In addition, it is based on biological evolution processes of living organisms derived from Darwin's theory of evolution, based on the survival theory of the fittest because of natural selection.

Genetic algorithms are a representation of a human belief that the human qualities are created with him and they are acquired by heredity. Many terms have been borrowed from genetics, for instance, (Population), (Mutation) and (Crossover). In these algorithms, we are trying to access the suitable solution, which depends on the principle of natural selection which based on keep good advantages in parents' generation to be transferred to children's generation aiming to get a new generation which has the best qualities and this is known as "survival of the fittest".

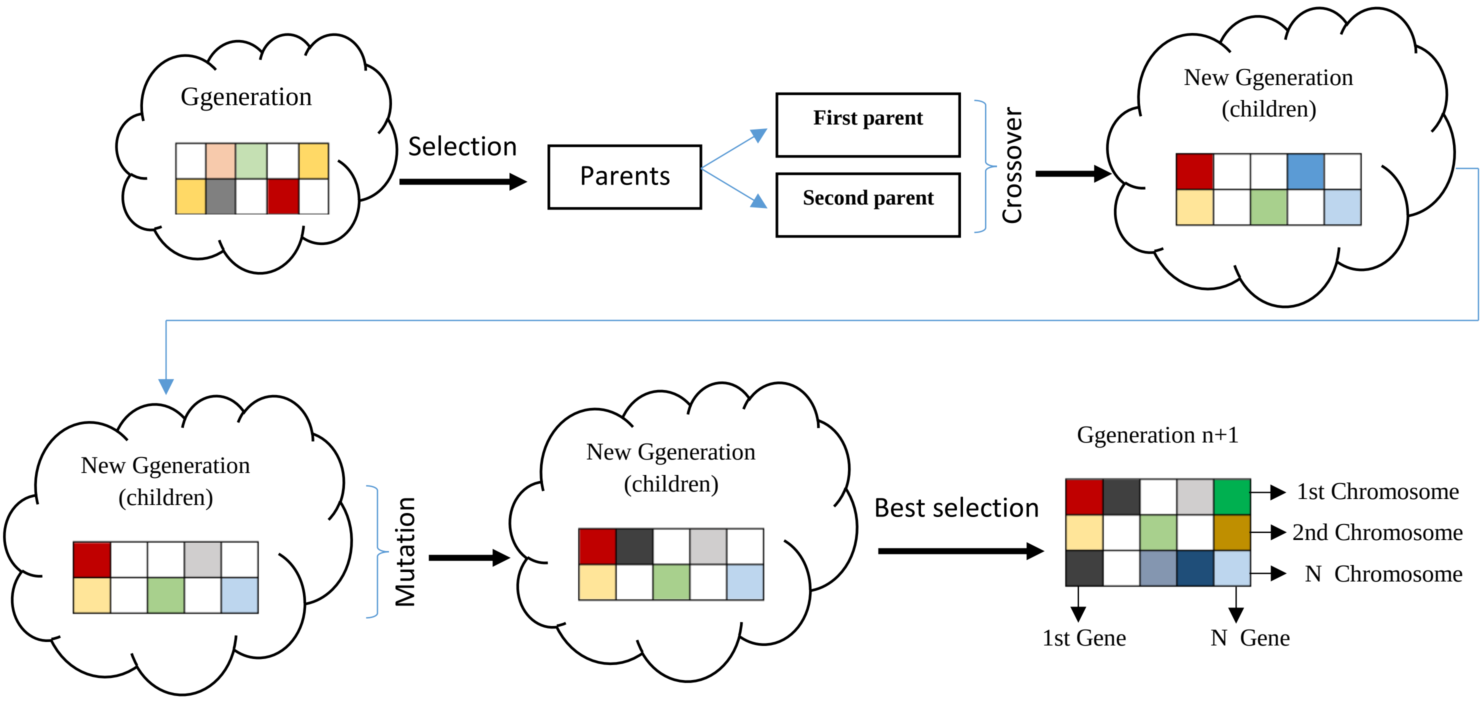


Figure 2: View to explain the basics of the work of the genetic algorithm

The effectiveness of the genetic algorithms in the fields where the research space is very large and complex and incomprehensible. In addition to the lack of methods of mathematical analysis to solve a problem or the failure of traditional research methods in finding solutions, and in the case of information or knowledge is scarce and difficult to represent the experience in this field. Consequently, this algorithm is often stronger, more effective in a random search.

Some of the problems which genetic algorithms are used in:

- **General examples of issues:**

  Numerical examples and calculations such as Travelling Salesman Problem (TSP). Industrial design such as the issue of wood cutting machine and scheduling shopping business also the examples of the quality of audio and video.

- **Automatic programming:**

  Genetic algorithms were used to improve computer programs in order to execute specific tasks and design another computer structure as sorting network.

- **Robots and machines:**

  Genetic algorithms were used in lots of automatic applications "machine –learning" including classification and prediction. In addition, they were used in neural networks design.

- **Economic models:**

  Genetic algorithms were used to model innovated machines, develop bidding strategies and also in the emergence of economic markets.

- **The interaction between learning and improving:**

  Has been used to study the mutual influence between individuals' learning and the evolutions of species.

- **Genetic algorithms are used as models of social systems:**

  They have been used to study the evolution aspects of social systems such as evolution of cooperation, the evolution of communication and trail-following behaviour in ants.

Genetic algorithms can be used in a large number of applications as well as in business, games and robots. There are a lot of other fields which have been used by genetic algorithms and it is difficult to mention all of them in this research.

From the above, genetic algorithms are one of the optimization algorithms (OAs), which is one of the techniques of artificial intelligence (AI) that works to find the best solutions in a smart way, reversing the traditional methods that are difficult to find the best solutions in many cases.

2.4.1 The basic principles of GA

The basic principles of Genetic Algorithms (GAs) were introduced by Holland (1975). Elhaddad (2009) described GAs In general as follows:

"Genetic algorithms start with generating random populations of possible solutions. Each individual of the population is represented (coded) by a DNA string, called a chromosome, and the chromosome contains a string of problem parameters. Individuals from the population are selected based on their fitness values. The selected parents are recombined to form a new generation. This process is repeated until some termination condition is met".

General algorithms for genetic algorithms:

```
BEGIN GA
    Initial a population of solutions.
    Evaluate the individuals in the population
        While NOT stop DO       // condition is not reached
            BEGIN
                Select parents from the population through a selection.
                Produce children from the selected parents.
                Crossover the parents to create offspring.
                Apply mutate the individuals.
                Evaluate fitness of chromosome.
                Extend the population adding the children to it.
            END
    Output the best individual found.
END GA
```

Figure 3: Abstract genetic algorithm

Note: the stop condition is not constant; it changes from problem to another.

The condition may be according to the (number of generations, a specific time for stopping, choosing the best solution, etc.). Moreover, the stop condition should specify depending on the problem.

2.4.2 Population Size

The first step in genetic algorithms is to create an Initial generation. Depending on the problem, the constraints on the initial generation can be determined, but the initial generation may be random and does not take into consideration any constraints.

The size of the population in the generation must be based on the techniques used in the algorithm, different large sizes are optimal. The larger the population, the greater the exploration of the global search space and the more time spent compared to whether the size of the population is small.

2.4.3 Encoding of chromosomes

Chromosomes are a representation of solving a particular problem to apply genetic algorithms. The solutions of the problem should be characterised to represent one of the chromosomes' encoding ways. The representation process of solving the problem is done by one of representing ways and there are lots of these ways. It must take into account that each chromosome or genetic is an expression of a small part of the solution, and any change in the site of any chromosome will affect the process and it will not give the original solution, but we will get a new solution due to the site of the genetic or chromosome in the representation series which represent the characteristics and the qualities of solving the problem.

Thereafter, a set of mathematically derived operations is used by biological processes to get a set of chromosomes in the end, which represents the final generation and each chromosome is only an individual of generation members. The best chromosome is the best solution to the problem at hand. To solve this problem by the software we should encode every chromosome to facilitate the work. Good coding may be the most important factor for the performance of any genetic. In many cases, the problems to be encoded can seem more easily represented by data types other than binary (Anderson and Simpson, 1996). There are many ways to represent these solutions by using genetic algorithms and the most important ways of them are:

2.4.3.1 Binary representation

Each chromosome is represented as a group of (1,0). To encode the entire chromosome, binary encoding is profusely used because it is easy to deal with, which each chromosome represent a set of zeros and successive units. In addition, this type is one of the most famous representation types and the reason behind its fame is because it was the first technique which was followed in the encoding process of chromosomes in genetic algorithms. Also, there are many issues in this type that have been being used to find the solutions.

Figure 4 An example of binary encoding:

Chromosomes 1	**1 1 0 0 0 1 0 1 1 1**
Chromosomes 2	**1 1 1 0 1 1 1 0 0 0**

Figure 4: Binary encoding

2.4.3.2 Digital representation

As it can be represented by using the decimal system, it also called (Digital Encoding). To represent chromosomes in this type, we have to use numerical values. It must take into account that there is a link among them and the problem to be solved. These values should reflect the solutions of the problem.

Chromosomes 1	**8 3 6 4 7 2 9 5 1**
Chromosomes 2	**2 5 4 7 1 9 6 8 3**

Figure 5: Digital encoding

2.4.3.3 String representation

Chromosomes can be represented as textual or literally (String Encoding). The representation process of chromosomes is done by using letters, words, texts or symbols to express the problem which we want to find a solution for it. This representation should be closely linked to the problem. Also, this type could be profusely used in medicine because they need to analyse the types of diseases; for instance, determine blood types, classify diabetics and hepatitis virus.

In this type, we need to develop new ways of crossover and mutation so as to fit with kinds of representation.

Chromosomes 1	**Right left Forward backward**
Chromosomes 2	**A+ A- O+ O- AB- AB+**

Figure 6: Textual and literally encoding

2.4.3.4 Hybrid representation

This type is the last one of representation types. It is a representation of the previous methods (Hybrid Encoding). Chromosomes could be represented by using a set of types that mentioned previously according to our needs. This type could be used if the previous methods cannot find a suitable encode to solve the problem.

2.4.4 Selection operators

The stage of selection is the first phase in the genetic algorithms. In this stage, a new generation is selected from the old one and all the chromosomes are put into groups. The right ones are chosen according to appropriate type by the objective and unbiased way.

Due to the presence of many election procedures, we have to choose the suitable procedure because of its impact on algorithms and get the best solution as soon as possible. There are many ways of selection but we will only explain the important ones:

2.4.4.1 Roulette Wheel Selection

The proposal basics of this method of selection by Holland, J. (1975). The most simple rule used in family selection is by the roulette wheel selection. To prosecute this type we should bear in mind that there is a roulette wheel that had been divided into 100 sectors and generation members are distributed on these sectors according to the median probability of election for each member of the current generation which gives the following mathematical relationship:

$$P_{select}(i) = \frac{F_i}{\sum_{j=1}^{n} F_j}$$

Where:-

i **possibility of individual election :** $P_{select}(i)$

Individual scale i F_i

number of generation members : n

Election processes are done randomly in move the wheel's as shown in the *Figure 7* and wait until the wheel stops on index then it takes the individual which pointed on it.

The more appropriate chromosome's degree as shown in the *Table 1*, it increases the number of its sectors. In other words, chromosomes are put in segments which correspond to the validity of their values, where the chromosome which has high validity occupies the largest sectors. In that case, it is more likely to elect a chromosome due to the occupation of many sectors. This means that the chromosome impacts positively on the next or current generation. The wheel will remain as the original and will spin again until all new population is filled up.

Table 1: Appropriate scale of chromosome in roulette wheel selection

	Appropriate scale F	Possibility of selection
First chromosome	13	0.11
Second chromosome	22	0.19
Third chromosome	15	0.13
Fourth chromosome	19	0.17
Fifth chromosome	25	0.22
Sixth chromosome	16	0.14
Seventh chromosome	5	0.04
Total	**115**	**1.00**

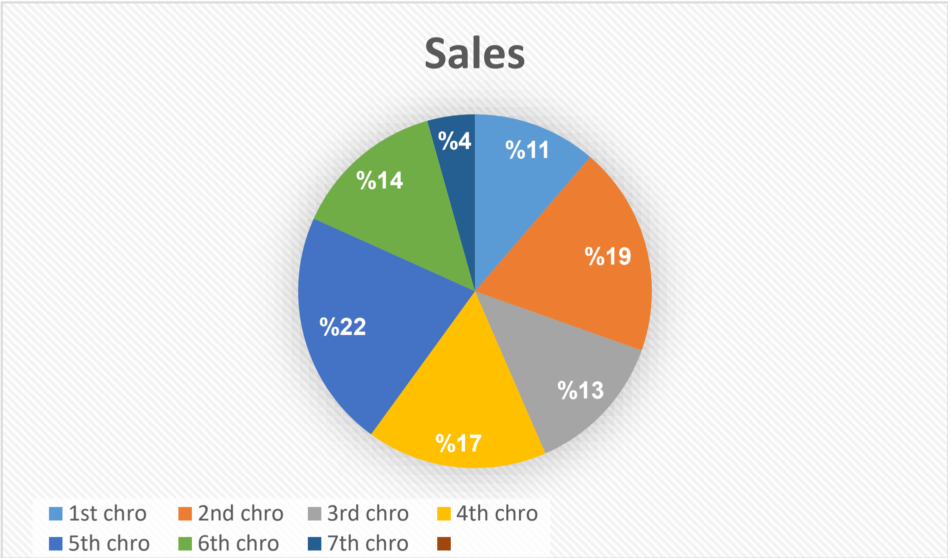


Figure 7: The selection process of roulette wheel

2.4.4.2 Elitism selection

The selection of this type should not ignore the good solutions in the current generation during the process of creating a new generation. From this logic, good solutions should be retained. Elitism way has been found which depend on copying the good solutions to take advantage of them in the next generations.

2.4.4.3 Tournament selection

Tournament selection is one of many methods of selection used in GA. In this operation a tournament between two random individuals from the population is held and the winner is selected (the one with the best fitness). Tournament selection proposed by Rich (1995).

2.4.4.4 Best selection

The best chromosomes are selected from the generation so that must calculate the value of quality function or fitness function to find out what is the appropriate extent of each chromosome and the best chromosomes are chosen. In case that there are chromosomes have same quality they should be chosen randomly and this just in case of equality.

2.4.4.5 Random Selection

Chromosomes are chosen randomly to be transferred to the new generation regardless the quality value which appropriates to chromosomes.

2.4.5 Crossover Operators

The crossover process is one of the most important processes in genetic algorithms which simulate the biological process. Next to parental choice (Representing two series of numbers or character) apart of these series are exchanged randomly. Thus, the population evolution from one generation to another that follows. There are many types of crossover chromosomes but we will only present some of them:

2.4.5.1 Classical crossover

The Classical crossover was suggested by Holland, J (1975). This process is done by choosing a cross point randomly. The chromosome is divided into two equal parts, and then these parts are exchanged with each other to give us a new generation to improved offspring.

Figure 8 illustrates how to apply classical crossover on two consecutive chromosomes of parents' generation according to random values.

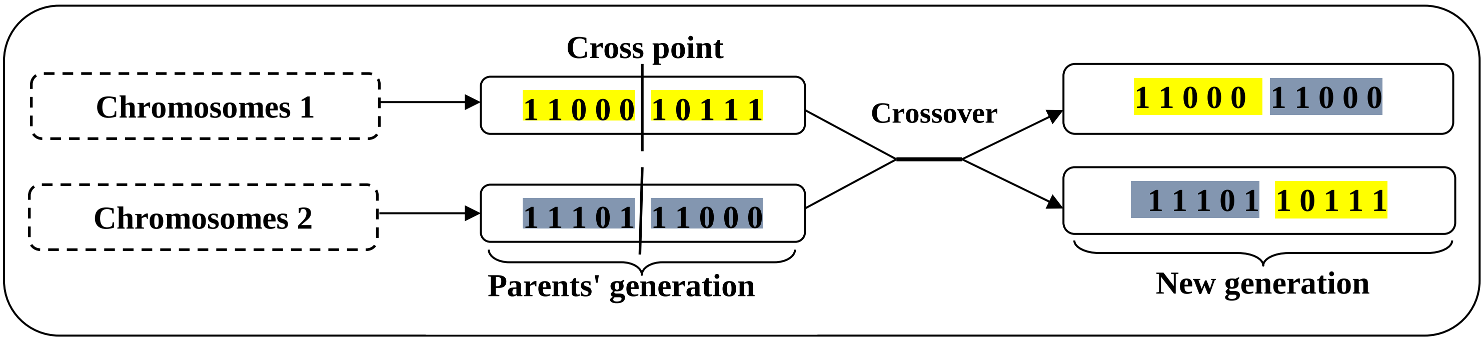


Figure 8: Classical crossover

2.4.5.2 Partially - Matched crossover (PMX)

The partially mapped crossover was proposed by Gold-berg and Lingle (1985). PMX can be viewed as an extension of two-point crossover. It is selected randomly from the parent's chromosomes to produce the offspring plus a repairing procedure, where some traits and values are passed from the generation of parents to children without any change in the genetic traits. Considering the change of the rest of the genetic traits in the chromosome. PMX works as follows:

<u>Procedure: PMX</u>

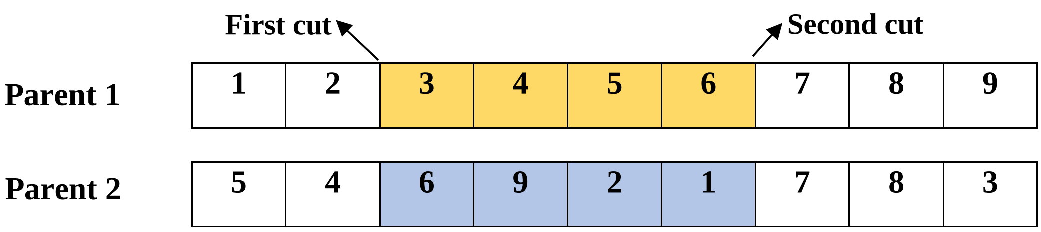
Step 1: Select two positions along the string uniformly at random. The sub strings defined by the two positions are called the mapping sections.

Step 2: Exchange two sub strings between parents to produce proto-children.

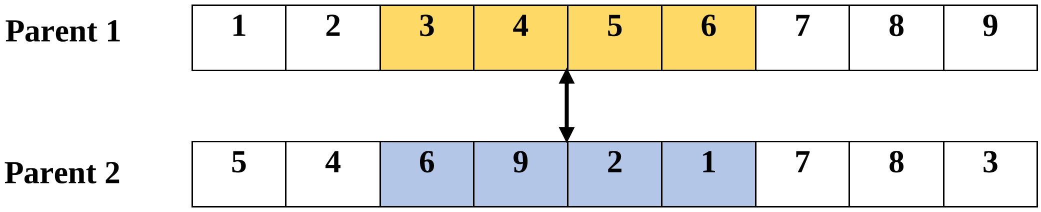
Step 3: Determine the mapping relationship between two mapping sections.

Step 4: Legalize offspring with the mapping relationship.

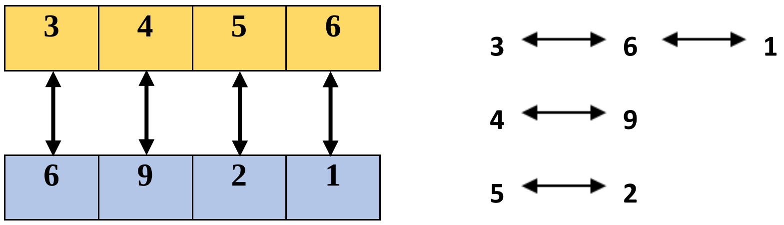
1. Select the sub string at random by identified two cut points



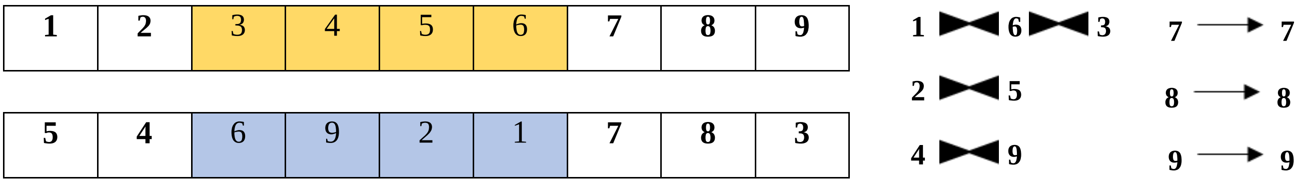
2. Exchange sub strings between parents



3. Determine mapping relationship



4. Legalize offspring with mapping relationship



After crossover

Offspring 1	6	5	6	9	2	1	7	8	4
Offspring 2	2	9	3	4	5	6	7	8	6

Figure 9: Partially - Matched crossover

2.4.5.3 Cycle crossover (CX)

This type was suggested by Oliver et al (1987). It is designed to create an offspring from the parents, where two cycles are performed in each cycle. Elements of one parent are chosen to create half of the child. CX works as follows:

<u>Procedure: CX</u>

Step 1: Find the cycle which is defined by the corresponding positions of cities between parents.

Step 2: Copy the cities in the cycle to a child with the corresponding positions of one parent.

Step 3: Determine the remaining cities for the child by deleting those cities which are already in the cycle from the other parent.

Step 4: Fulfill the child with the remaining cities.

1. Find the cycle defined by parents.

Parent 1	**1**	**2**	3	4	5	6	**7**	**8**	**9**
Parent 2	**5**	**4**	6	9	2	3	**7**	**8**	**1**

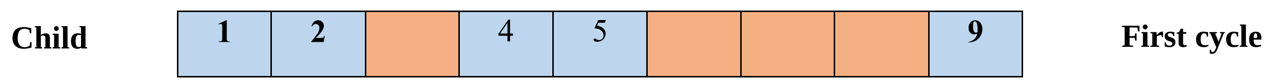
Cycle **1 → 5 → 2 → 4 → 9 → 1**

2. Copy the cities in the cycle to a child.

Child	**1**	**2**		4	5				**9**
Parent 2	**5**	**4**	6	9	2	3	**7**	**8**	**1**

First cycle

3. Determine the remaining cities for the child.



4. Fulfill the child.

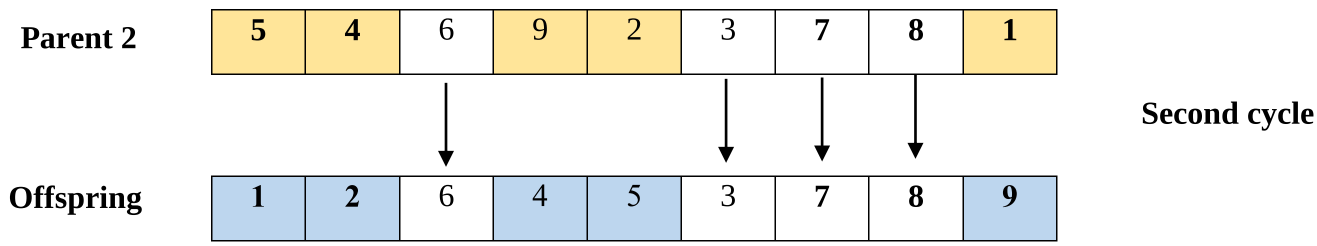


Figure 10: Cycle crossover

2.4.5.4 Order crossover (OX)

Order crossover was proposed by Davis (1985). It is a slight variation of position-based crossover in which the order of the cities in the selected position in one parent is imposed on the corresponding cities in the other parent. OX works as follows:

Procedure: OX

Step 1: Select a sub string from one parent at random.

Step 2: Produce a proto-child by copying the sub string into the corresponding positions of it.

Step 3: Delete the cities which are already in the sub string from the second parent. The resulted sequence of cities contains the cities that the proto-child needs.

Step 4: Place the cities into the unfixed positions of the proto-child from left to right according to the order of the sequence to produce an offspring.

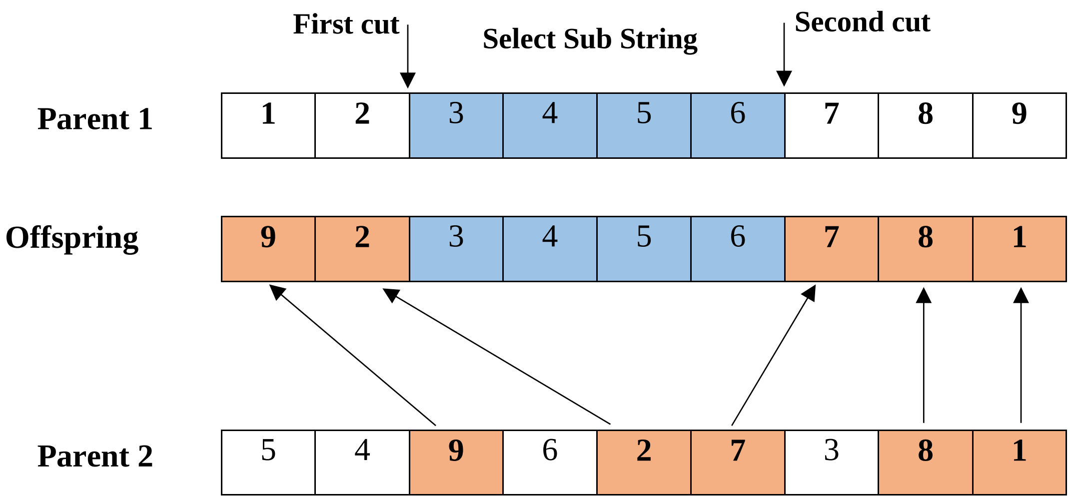


Figure 11: Order Crossover

2.4.5.5 MULTI-CROSSOVER TECHNIQUE

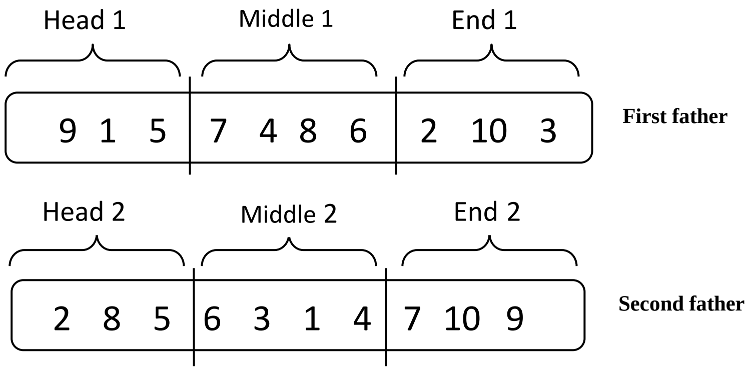
The multi-crossover technique was suggested by Mr Younis Alhadad and Mr Ayman Qunoos (2011). This technique is applied in the case that the primary generation consists of two individuals. On the contrary, the other crossover techniques need a huge number of individuals to apply it. In regard to multi-crossover technique needs only two individuals in the primary generation and from these two generations, we will get a new generation which its capacities are different from primary generation's capacities. The new generation consists of 104 new individuals. Around fifty different ways are applied in this technique. Each individual of the new generation inherits different attributes. The basic principle of this technique is used two cut points which are determined randomly. Assuming that two points are determined (p1, p2) we will get chromosome's head which is (1, 2, ..., p1 -1) and chromosome's middle (p1, p1+1, ..., p2). Finally, chromosome's end (p2+1, p2+2, ..., n).

For example, if we have the following two chromosomes:

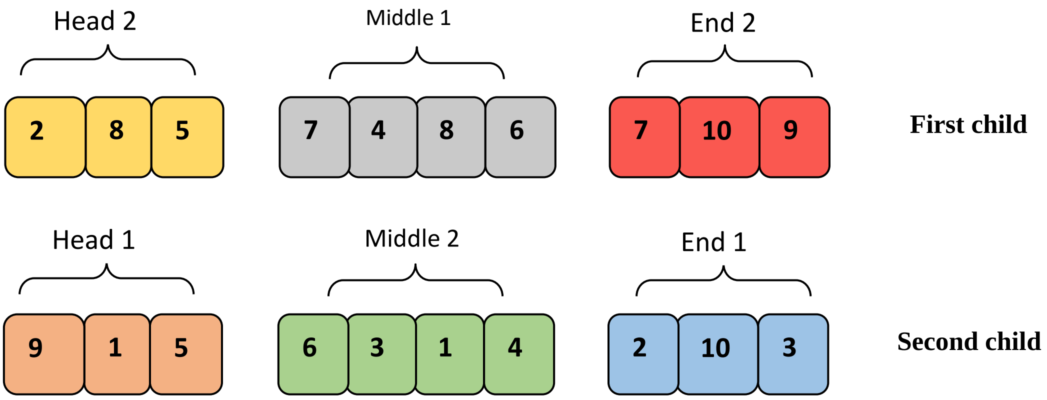
9	**1**	**5**	**7**	**4**	**8**	**6**	**2**	**10**	**3**

2	**8**	**5**	**6**	**3**	**1**	**4**	**7**	**10**	**9**

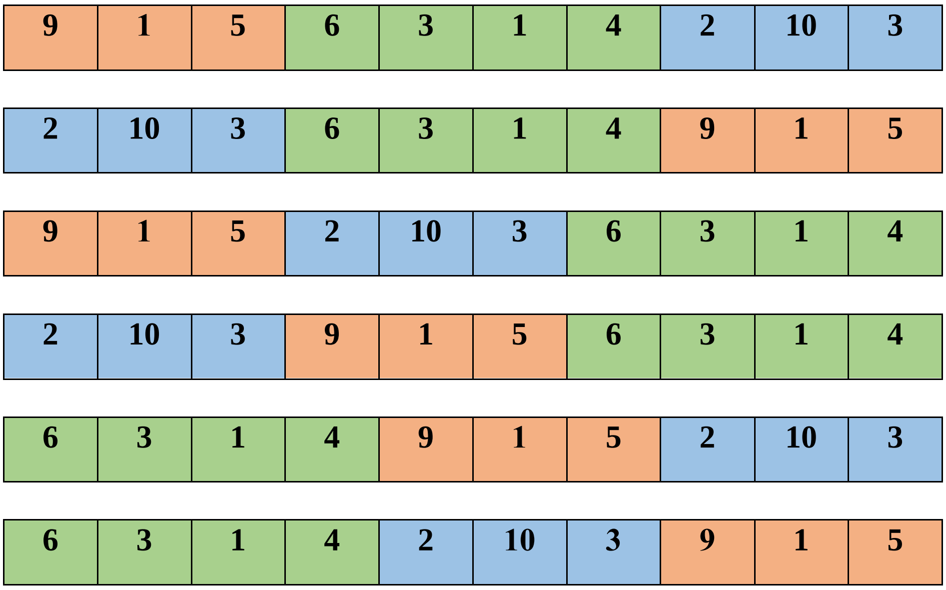
points of crossover are delimited where $P1 = 4$ and $P2 = 7$



After determining the crossing point, the previous sections will be exchanged with each other to produce a new generation and this is the **first step:**



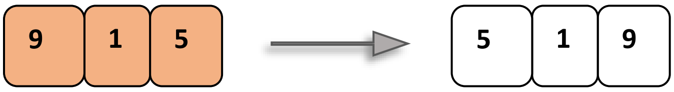
So, by this way, we will get twelve different children can be obtained **2 x (3!)**.



Six different children from (head1, mid2, tail1)

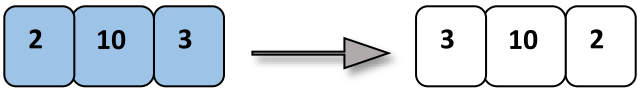
Figure 12: Multi-crossover technique (first step)

Second step: elements in head 1 and 2 are exchanged as:



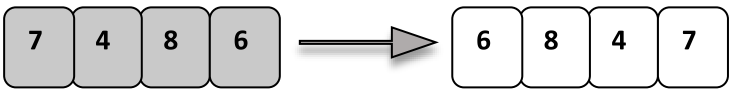
Next, we exchange the elements in head 2 (second father) then we repeat the **first step**, so we get new twelve children.

Third step: elements in end 1 and 2 are exchanged as:



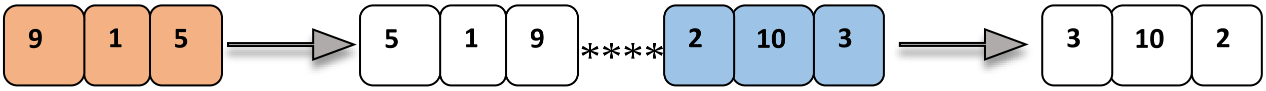
Next, we exchange the elements in end 2 (second father) then we repeat the **first step**, so we get new twelve children.

Fourth step: elements in middle 1 and 2 are exchanged as:



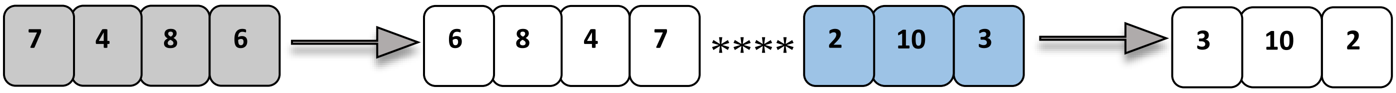
Next, we exchange the elements in middle 2 (second father) then we repeat the **first step**, so we get new twelve children.

Fifth step: elements in head 1, end 1, head 2 and end 2 are changed their order as:



Next, we change the elements in head 2 and end 2, after changing their order the **first step** should be fulfilled and the result will be new twelve children.

Sixth step: elements in middle 1, end 1, middle 2 and end 2 are exchanged as:



Moreover, this figure explains the process on middle 1 and end 2 and then fulfils the **first step** to getting new twelve children.

2.4.6 Mutation

Mutation is the last stage of iterative processes series which contribute to reaching the best solution quickly. It is derived from genetic algorithms which are a branch of random search techniques for the best solution, and it immediately comes after the crossover process.

Mutation is an operation which introduces small, random modifications to an individual according to mutation probability.

2.4.6.1 Classical Mutation

The Classical mutation was suggested by Holland, J (1975). This process can be described as a gene change within the chromosome. This process is important because it adds some new features which may not exist in the parent's generation.

Before mutation	Chromosomes 1	1	2	3	4	5	6	7	8
		↓	↓	↓	↓	↓	↓	↓	↓
After mutation	Chromosomes 1	1	2	3-	4	5	6	7	8

Figure 13: Classical Mutation

2.4.6.2 Displacement Mutation (DM)

Displacement mutation was presented by Michalewicz (1992). It was necessary to determine subtour randomly. Subtour is a group of genes within the chromosome; this group is transferred and put it randomly in another place. This type is also known as cut mutation.

For example, if we have:

Select a subtour at random

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Insert it in a random position

1	**4**	**5**	**6**	**2**	**3**	**7**	**8**	**9**

Figure 14: Displacement mutation

2.4.6.3 Exchange Mutation (EM)

Exchange mutation operator Banzhaf (1990). Exchange mutation selects two positions in the chromosome at random and then swaps the genes on these positions. Moreover, this type is known as a Swap Mutation.

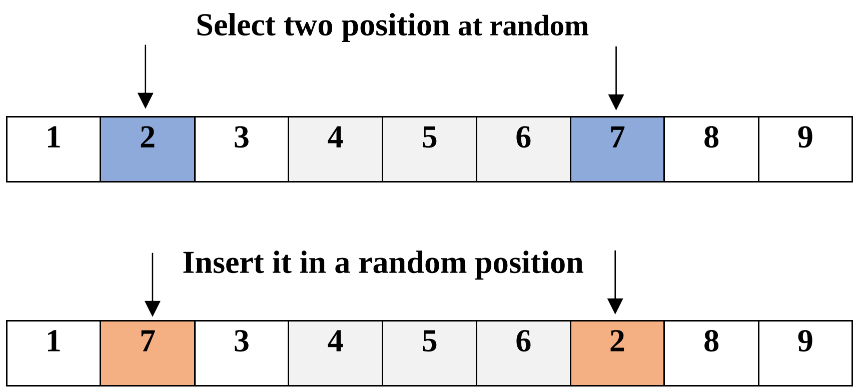


Figure 15: Exchange mutation

2.4.6.4 Insertion mutation (ISM)

Michalewicz(1992) suggested the basics Insertion mutation more than a 25 years ago. The operator initially selects two random crossover points in the parent string. The subtour between points is then selected and reversed (inverted). This is shown in the example below:

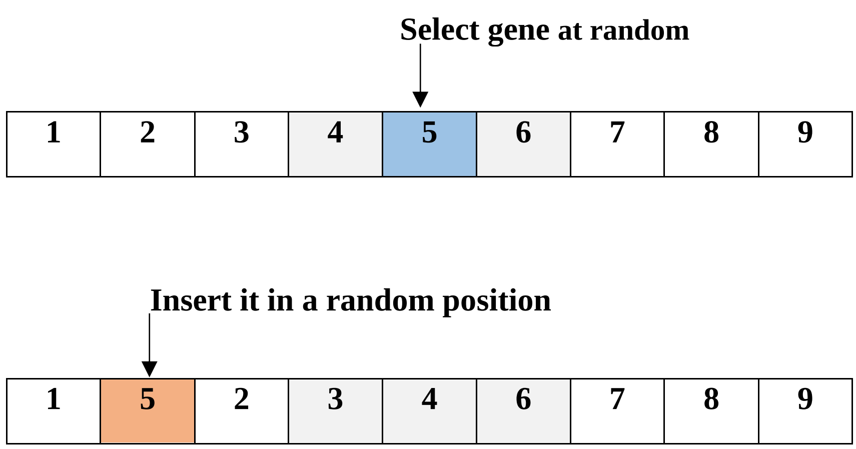


Figure 16: Insertion mutation

2.4.6.5 Heuristic Mutation (HM)

The heuristic mutation was proposed by Cheng and Gen (1996). It is designed with the possible permutations technique in order to produce an improved offspring. A set of chromosomes transformable from a given chromosome by exchanging no more than **N** of genes.

The best one among the neighbourhood is used as offspring produced by the mutation. The mutation operator works as follows:

<u>Procedure: Heuristic Mutation</u>

Step 1: Pick up **N** of genes at random.

Step 2: Generate neighbours according to all possible permutations of the selected genes.

Step 3: Evaluate all neighbours and select the best one as offspring.

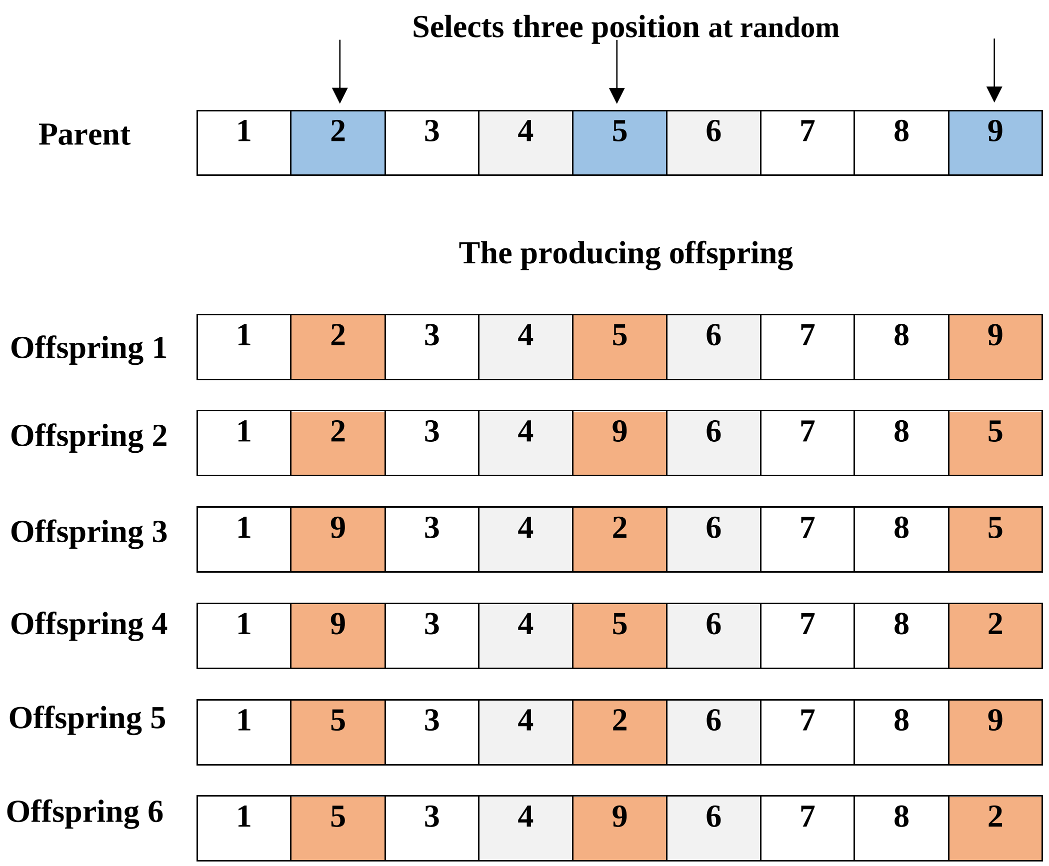


Figure 17: Heuristic mutation

2.4. 7 Fitness function

The fitness function is one of the most important molecules in the genetic algorithms and it is the main link between the problem to be solved and algorithms. This function takes each chromosome separately and evaluates its extent performance in solving the problems by giving it a specific value. Whenever this value is larger the chromosome is more efficient or the goal may be diminishing the value of effectiveness functions. Hence, chromosome which has less effectiveness function is the best and that according to be a problem to be solved.

Chromosomes that have high fitness have the opportunity to stay because they are more useful to solve the problem. Chromosomes' fitness increase by repeating these steps. When algorithms processes finished, chromosomes will reach the highest level of fitness and that reflects the good solutions to the problem.

As a result of the above, we should say that genetic algorithms are not linked to the problem to be solved except by the fitness function. Genetic algorithms can be used to solve many issues without rewriting or changes any part of algorithm expect fitness function and encoding the problem.

2.5 Graph theory

R.Balakrishnan et al (2012) Explain that the graph theory is:"Graphs serve as mathematical models to analyse many concrete real-world problems successfully. Certain problems in physics, chemistry, communication science, computer technology, genetics, psychology, sociology, and linguistics can be formulated as problems in graph theory. Also, many branches of mathematics, such as group theory, matrix theory, probability, and topology, have close connections with graph theory.

Some puzzles and several problems of a practical nature have been instrumental in the development of various topics in graph theory."

Problems of linear programming and operations research (such as traffic problems) can be tackled by the theory of flows in networks. Scheduling problems are examples of problems that can be solved by graph theory. The study of simplistic complexes can be associated with the study of graph theory. Much more such problems can be added to this list.

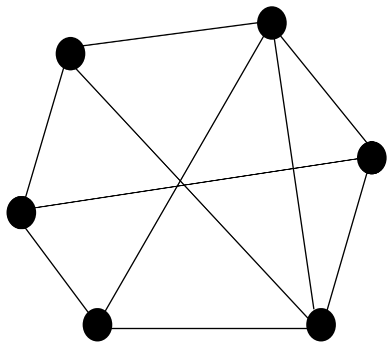


Figure 18: Graph theory

Chapter 3

Literature Review

In this chapter, we will discuss some of the related works have been achieved from different point view, hence, we discuss the techniques used from many researchers who obtained different results and compare those techniques with our technique.

The actual problem faced by researchers in this field is the lack of the research publications about the university timetable. Researchers have proposed a number of approaches to solving the problem of scheduling.

Previous literature that related to timetabling problems have been surveyed, and the focus was on the course and examination timetabling.

Several methods can be used to solve the timetabling problems, many paths or techniques or methods of encoding or representation can be used depending on the problem and its appropriate representation methods. A summary of previous studies will be presented below.

3.1 Course TimeTable

3.1.1 Description and Application of Genetic AlgorithmTimeTable

According to Min Wang (2012), the models presented include five elements: time, place, professor, lecture and class. The definition of each element has been given as below.

Time slot set: T = $\{t1,2,\cdots,ta,\cdots tA\}$.

T= 5 (days) × 4 (slots) = 20.

Room set: R = $\{r1,2,\cdots,rb,\cdots rB\}$

$N=T\times R=\{(t_1,r_1),(t_1,r_2),\cdots,(t_a,r_b),\cdots,(t_{20},r_B)\}$

Professor set: P = $\{p1,2,\cdots,pc,\cdots pC\}$

Lecture set: L = $\{l1,2,\cdots,ld,\cdots lD\}$

Class set: C = $\{c1,2,\cdots,ce,\cdots cE\}$

Min Wang (2012) presented in his thesis hard constraints on the course timetable, the constraints were as follows:

"H1: All allocated rooms are large enough .

H2: At the same time, each class cannot have a lecture in more than one room .

H3: No omission of classes in the timetable is demanded .

H4: At the same time, each professor only can teach one lecture in one classroom .

H5: At the same time, every room cannot be arranged for more than one lecture."

The author uses a two-dimensional matrix and used binary numbers as the element of the matrix, this is the initial population. Figure 40 shows representation of the matrix.

	(t_1, r_1)	(t_1, r_2)	$\cdots$	(t_a, r_b)	$\cdots$	(t_{20}, r_B)
c_1						
c_2						
$\cdots$						
c_i				$a_{i,j}$		
$\cdots$						
c_E						

Figure 19: Two-dimensional matrix model

All of the values of the parameters mentioned in this model are given in *Table 2*.

Table 2: The parameter list

Parameters	Value
Number of time-slots	20
Number of rooms	5
Number of professors	10
Number of classes	4
Number of lectures of each professor	4
Crossover rate	0.75
Mutation rate	0.02
Population	30

"As expected, the results will be converged and we obtain the best result which is around 650. And it can be observed; the higher population size will give the best result faster" Min Wang (2012).

Min Wang (2012) presented results in his thesis by comparing the time consumption with different population sizes as in *Table 3*.

Table 3: The result of comparison

Population	Time[Sec]
10	43.1193
20	88.9606
30	133.7512
40	179.7556
50	228.0954
60	269.2885
70	313.9937
80	362.2176
90	426.1943
100	520.9020

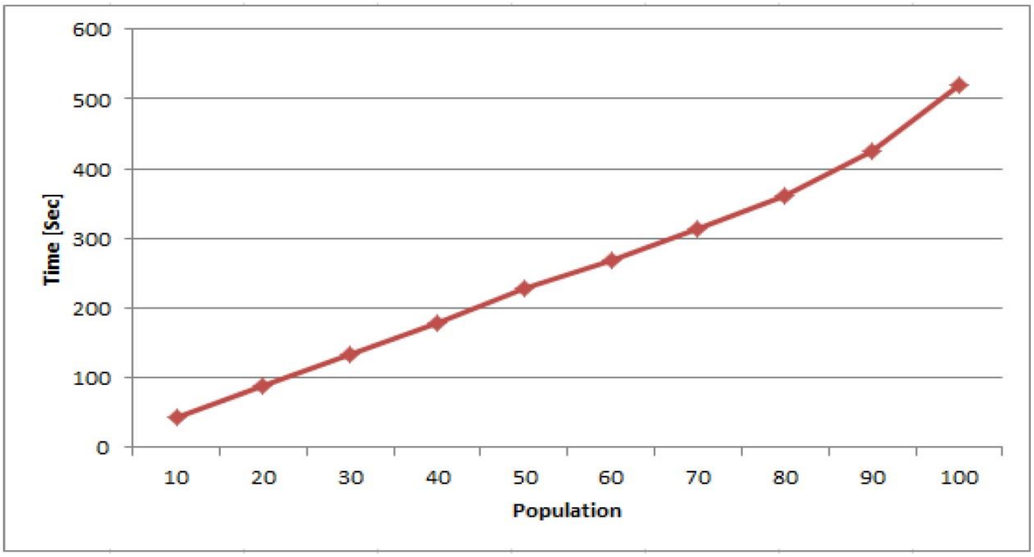


Figure 20: Time consumption of different population sizes

3.1.2 OPTIMIZATION OF ACADEMIC TIMETABLING USING GENETIC ALGORITHM

According to the work of Teoh Chong Keat (2012) which is about the academic timetabling and the use of genetic algorithms in solving the problem of its preparation. The academic timetabling problem considers a set of lecturers who are already pre-assigned to a course beforehand, a set of rooms and a set of time series data of the availability of each room for the entire week. Observance Resources available for the course timetable for the entire week.

Basically, the timetable can be divided into 5 study days. There are 8 useable time slots, regard not to use periods timeslots are used for recess and prayer.

Day \ time	**8-8:50**	**9-9:50**	**10-10:50**	**11-11:50**	**12-12:50**	**1-1:50**	**2-2:50**	**3-3:50**	**4-4:50**
Monday									
Tuesday									
Wednesday									
Thursday									
Friday									

Figure 21: Typical Timetable Model

In this research use representation is a string of bits with the position of the binary digits, 0 or 1 having a particular meaning when decoded. For this research, the chromosome is encoded by the set of classes or genome. The genomes encompass the entity, of course, day, room, and time.

Chromosome = {(course i **, day** i **, room** i **, time** i**)}**

where :- i = 1:N \ N = Number of Courses.

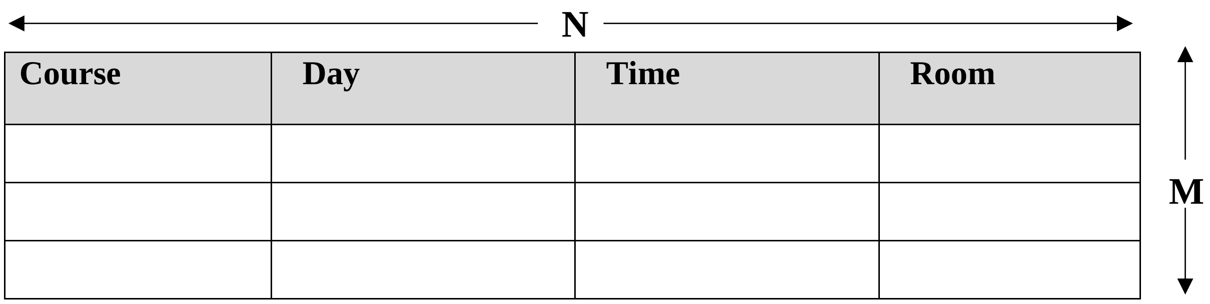


Figure 22: Chromosome Model

All of the data has to be represented by integer accurately to be used by the evaluation later. The fitness value has to be as near to zero as possible. The constraints are given as follows:

i. A Room can only have 1 Subject at 1 time

ii. No Subject on Monday till Thursday between 1.00pm-1.50pm

iii. No Subject on Friday between 1.00pm-2.50pm

"In this research, the experiment is repeated for three times and the average fitness score is averaged and tabulated into *Table 4*" presented by Teoh Chong Keat (2012).

Elapsed Time: Seconds.

Table 4: Elapsed Time Table

No of Iterations / No of course	50	100	500	1000
50	72.08	147.08	726.42	1426.28
100	240.93	506.40	4268.21	4550.91
150	1085.23	974.10	4708.54	21863.58
200	1384.45	1610.59	8050.29	16192.53
250	1218.93	2504.61	12440.28	24534.77

"In this research, the accuracy of the timetable for 50, 100, 150, 200 and 250 courses were tested and the accuracy percentage yielded 98.5%, 92.0%, 89.6%, 85.3% and 82.5% respectively. The result and the mean elapsed time were tabulated in *Table 5* below" Teoh Chong Keat (2012).

Table 5: Result Table

No.	No. of Courses	Accuracy (%)	Mean Elapsed Time (secs)
1	50	98.5	592.9
2	100	92.0	2391.6
3	150	89.6	7157.9
4	200	85.3	6809.5
5	250	82.5	10174.6

The best results were from the previous thesis:

No. of Iterations: 100

No. of Courses: 50

Fitness Value: 0

3.1.3 Solving timetable problem using genetic algorithm

Walid Y. (2011) has a proposed and implement the algorithm using genetic algorithms to solve timetable problem for the closed yearly system in one of the colleges of University of Benghazi. In addition to the above, the algorithm that was designed was based on the preparation of a timetable of the student depends on the wishes of students through the questionnaires to determine the requirements of the table and to reflect later on the fitness function as hard and soft constraints.

The action of the fitness function focuses on a set of points (Number of lectures in one day for a student, Time of lecture, Break between lectures and Day off for student).

Reference to provide by Walid Y. (2011) in his thesis. Was used a table for represents a timetable problem thus every cell in the table appear a lecture. Applied a hybrid representation type was used to represent lecture data from course number, teacher number, and classroom number. Figure 44 shows how Walid Y. (2011) represents data. Where they represent:

(CS2, MSE, MIT) represents the subject.

(WG003, WG607) represents a lecture.

(CC2, CC3, CC1) represents classroom.

P1	P2	P3		……	……	Pn
MSE,WG607,CC2			MIT,WG607,CC1			
			MSE,WG607,CC2			
			CS2,WG300,CC1			
MSE,WG607,CC3						

Figure 23: Lecture representation table

Table 6 shows a sample of the results obtained by conducting experiments in a thesis submitted by Walid Y. (2011).

Table 6: Result of population size evaluations

No.	Population	Time[Sec]	Fitness function
1	25	00:23	16838
2	50	00:50	17373
3	100	02:01	17673

3.2 Exams TimeTable

The difference between course timetable and exam timetable makes the results very different according to the time and the Iterations to reach the best solution. Thus, we compare the results reached in this thesis with results of some previous literature even that we can determine the importance of what has been achieved.

3.2.1 Exam Timetabling Problem Using G.A.

Twinkle Dahiya et.al (2015) offered a solution to the problem of examination schedules, where the results of his work were as in Figure 24.

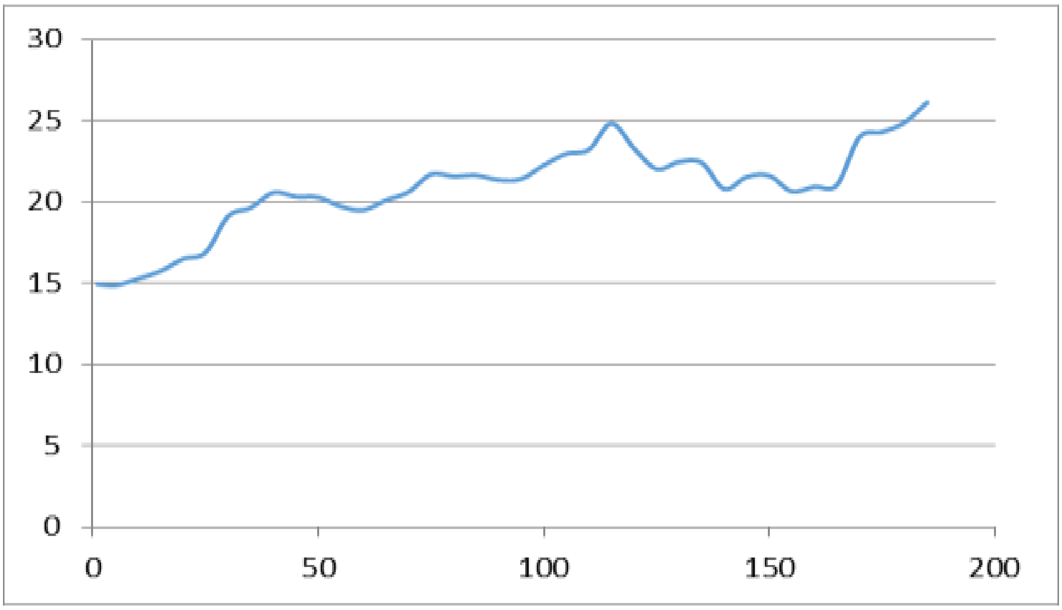


Figure 24: Generation vs. Total fit

According to Twinkle Dahiya et.al (2015), " The process is repeated over many generations. Some of the generated parent chromosomes which represent the clash-free timetable with their fitness values for generations. Growth graph includes the variation of mean fitness value and total fitness value from generation 1 to generation 150. The following graph shows the comparison of a generation with Total fitness values. The horizontal axis shows Generation number with a gap of 50. The vertical axis shows Total fitness value with a gap of 5.

The following graph shows the comparison of a generation with total fitness values. We observe that at various instants the value increases and at some instants mean fitness value decreases. From generation 1 to generation 180, the mean fitness value increases from 13.9546 to 26.1058 i.e. 45% increase".

3.2.2 Secondary School Examination Timetabling using Genetic Algorithm

According to Azman Yasin et al. (2007), through the analysis of other researchers, it turned out that there are results, but these results only rarely timetables were produced successfully without any problem. The default parameters set are shown in *Table 7*.

Table 7: Default parameters setting

Parameters setting	**Variable**
Slot Number	17
Population Size	40
Elitism	True
Crossover Rate	0.8 (80%)
Mutation Rate	0.05 (5%)
Selection type	Roulette Wheel
Maximum Generation	1000

Table 8: Results T8 and from T16 to T20

Number of Testing	**Population Size**	**Best Generation**	**Time (Sec.)**	**Fitness Score**
T8	100	10	0.406	2770.0
T16	20	20	0.484	2660.0
T17	40	60	0.203	2580.0
T18	60	32	0.890	2820.0
T19	80	11	0.547	2660.0
T20	100	22	0.953	2670.0

Azman Yasin et.al(2007) mention that "Each testing with a different population size produces a good result. The assumption on the small population sizes can provide a quick calculation and less time requires in generating the data. From the results, T17 is still the best solution without clash and consecutive. As the test whose settings or parameters are the default settings of the system. In the meantime, T16 and T19 were the acceptable solutions without clash and consecutive. More time is taken to produce others generation if compared with the time taken to produce T8 generation".

3.2.3 A GA Approach for a Real-World University Examination Timetabling Problem

Oluwasefunmi T. et.al(2010) provided in his research paper solved the problem of preparing examination timetable. "In general we would think of a good quality timetable as one that is (firstly) feasible and that (secondly) satisfies the hard and soft constraints to an acceptable level".

Has tried to formulate the scheduling problem in a general way for taking into consideration all requirements for the examination schedule. Thus, we have the following datasets:

A set H = {h1,…, hn} of halls. A hall is a venue of specific capacity.

A set C = {c1,…, cm} of courses.

A set K = {k1,…, kk} of classes. A class is a group of students writing the same exam.

A set A = {a1,…,al } of exams. An exam is an instance of a list of halls, a list of courses and a list of classes. Each exam has a duration expressed in time slots.

A set T = {t1,...,tp} of time slots. Time slots are distributed in d weekdays and h daily periods.

K = (kij) AxA is the conflict matrix where each element denoted by Kij, (i,j e{1,…, L} is the number of classes taking exams Ai and Aj.

Through experiments that were presented by Oluwasefunmi T. et.al(2010) the total number of examinations is 437 exams with 8,000 students, and the number of exam days is 22 days. The total number of the hall is 42.

The algorithm was implemented using Java and the program used all data which has already been represented. The output of this system is quite large, therefore only a phase is shown in Figure 25.

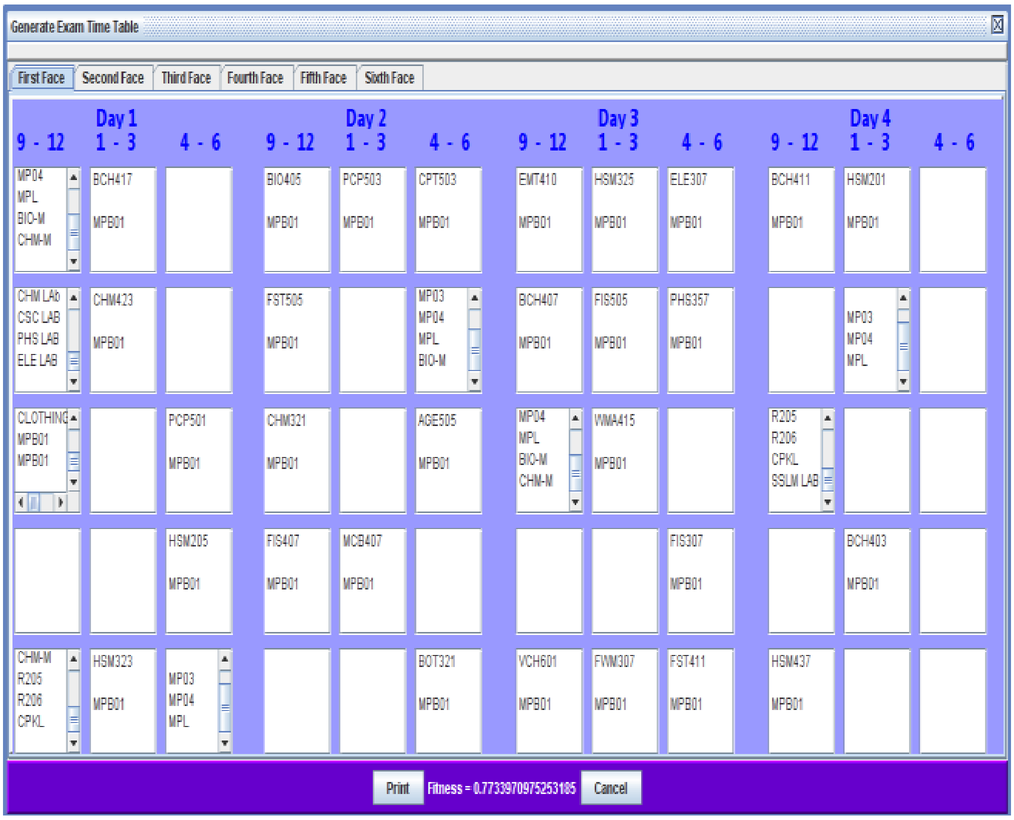


Figure 25: Sample of Examination Timetable Generated

Chapter 4

The proposed algorithm to solve Timetabling problem

4.1 Introduction

In the previous chapter**,** we focused on the basic concepts to solve the timetable (exams and lectures) problems. Moreover, the issues related to timetable formatting and the difficulties, restrictions, and the necessary requirements to accomplish a timetable which free of conflicts that meet the needs and requirements.

In this chapter, we show the design and implementation of an application that finds solutions to the timetable preparation problem in IT department at Benghazi University through executing the new proposed algorithm. This algorithm is different from the general algorithms in producing a new solutions as the following:

1. Lectures timetable.
2. Exams timetable.
3. Coding and decoding information and reporting.

The first two algorithms will be executed by using MATLAB 2013, and the third algorithm for decode information and assembling it in a report (timetable) for printing, executed using Visual Studio 2013.

4.2 The Techniques used to solve the problem

In order to get precise results, we used reliable techniques to solve the academic timetable preparation problem, one of them is proposed by Younis R. Elhaddad et.al (2010) which is has the ability to retrieve results in less time, and less consumption of computer resources during the implementation. The graph theory will also be used in the process of evaluating the quality of the solutions, for the ease of representation and the effectiveness in calculating and evaluating the solutions that result from the implementation of the algorithm.

4.3 The used Programming languages.

For the implementation, we use MATLAB for writing the algorithm's code. The decision for choosing this language is for its ease of use in formation and modification of matrices depending on our needs, and execution of the processes mentioned earlier, and then allowing us to execute the proposed solution more effectively.

In addition, we use Visual Basic to design a system that works on decoding information and reporting it in the tabular format. The reason for choosing Visual Basic language is its flexibility in dealing with reports and presentation of the results, and the ease of using the language's environment to accomplish a coding and decoding system.

4.4 The Time Table requirements

There are many methods can be used to represent the solutions, therefore; the relationship between genetic algorithms and the problem is the representation and fitness function (rating function). In addition, because there are many different ways for representation, the use of appropriate representation depends on the type of problem.

We have worked on the proposed algorithm to represent the restrictions as part of the timetable, which is easy and simple to deal with the table. Moreover, there will be no need to represent time, as every lecture will be dealt with independently.

To start coding process, we have to choose an effective way to represent the coding, in this research, we have chosen (the decimal representation) for timetable coding. After specifying the type of coding, an analysis of the timetable contents should be done to identify the elements that will be coded because different contents, and different coding is used. For example (***Classrooms***, ***Faculty members***, ..., etc.). The coding process is described in the following sections:

4.4.1 Classrooms

The data of classrooms that can be booked during the semester and its capacity must be specified. Each classroom should be given a unique number, which refers to the classroom.

Table 9: Classrooms list

Classroom Key value	Classroom Name	Capacity
1	C1	35
2	Hall2	70
3	C5	40
4	L3	30
5	C20	40
….	….	….
N	L1	25

4.4.2 Faculty members

The faculty staff members data are specified, each faculty member should assigned to unique number.

Table 10: Lecturers list

Lecturer Key Value	Lecturer Name
1	Ali M.
2	Salem R.
3	Omar S.
5	Hietham
6	Wafa
.	.
.	.
N	Waled F.

4.4.3 Courses

All courses and the number of groups in each course should be determined, in addition, estimation of the expected number of students, which registers for the course taking into consideration the lecturer group. For example, More than 100 students are expected to enroll in the course **IT111**. Students who are expected to enroll in the course must be divided into several groups. Assuming that the policy of the college or university that each group should not exceed the number of students in the 30 students per group, thus; we need four groups of the course and for other courses until the completion of all the courses that will be taught during the semester.

Table 11: Course - lecturer list

Course Key Value	Course Name	Course Code	Lecturer	Approximate students no.
1	Programming 1 (A)	IT111 (A)	Soulaf	50
2	Programming 1 (B)	IT111 (A)	Wafa	50
3	Programming II (A)	IT112 (A)	Amina	50
4	Programming II (B)	IT112 (B)	Ali	50
5	Discrete structures	IT201	Maha	100
6	Data structures (A)	IT212 (A)	Salem	40
7	Data structures (B)	IT212 (B)	Omar	20
8	Intelligent system	CS451	Mohamed	30
9	Compiler construction	CS431	Salem	20
.	....	....	....	....
.	....	....	....	....
N	Numerical Methods	CS205	Hietham	15

4.4.4 Time slots

The selection and division of the time slots available should be done according to the university or faculty policy, by representing the slot (from -- to --) where two spaces or numbers are given to represent the time slot.

Table 12: Time-slots

Key value	Period slot
1	8:00 - 9:30
2	9:30 - 11:00
3	11:00 - 12:30
4	12:30 - 2:00
5	2:00 - 3:30
6	3:30 - 5:00
7	5:00 - 6:30

After the process of filling the courses table with the lecturers, a connection must be made between courses, lecturers, and the classrooms.

Must fill the main inputs matrix, containing six columns. The first column is used for the course key values because each course will be taught twice during the week, there will be twice as many classes. The second column contains all the lectures to be taught during the semester, taking into consideration that, if any subject is taught more than one group of students, each group is treated as a single course. For example, if the course IT101 has to be taught to three different groups, such as; Group A, group B and Group C, then IT101A, IT101B, and IT101C will be considered as different courses as in *Table 13*.

Moreover, because each subject will be taught in two different time slots during the week, each one will be denoted with a different number. The reason for applying this technique stems from the fitness function design, which will be explained later. The third column shows the names of lecturers teaching the courses, with the classroom booked for the lecture given in the fourth column. The fifth column indicates the year to which the course corresponds, and the sixth column is used to indicate the similarity between the subjects, whereby the same subjects indicates by the same number.

Table 13: Main inputs matrix

No.	Course Name	Course Code	Lecturer	Classroom	Course year	Similarity
1	Programming 1 (A)	IT111 (A)	Hietham	C1	First year	1
2	Programming 1 (B)	IT111 (B)	Elsharri	C1	First year	1
3	Discrete Structures (A)	IT201 (A)	Ali	Hall7	Second Year	2
4	Data structures (A)	IT212 (A)	Omar	C3	Second Year	3
5	Data structures (B)	IT212 (B)		C11	Third Year	3
6	Intelligent system	CS451	Hietham	C2	Fourth year	4
7	Compiler construction	CS431	Younis	L1	Fourth year	5
.	....	....	....	....	....	....
.	....	....	....	....	....	....
.	IS Theory and Practices	IS321	Fatima	C3	Third Year	.
.	Network operating systems Software	CN312	Salah	NetL	Third Year	.
.	Software Project Management	SE492	Mohamed	C17	Fourth year	.
.	Image Processing	CE416	Kenz	CSDL	Fourth year	.
N	Numerical Methods	CS205	Soulaf	15	Third Year	N

After the process of creating a table and connecting it with the classroom table, the next step is a representation of the timetable. Each row in the *Table 13*, represents the course name, the lecturer, and the lecture place (classroom, lecture hall, or lab).

4.5 How to represent timetables

4.5.1 Representing TimeTables

In this thesis, we will present two ways to represent the timetable. Hence, we will use the easiest and most appropriate method at each stage according to the need to deal with it. The details about the two methods as follows:

One-dimensional matrix:

The digital representation will be used to represent the course in the timetable. The reason for choosing this type of representation is its ease of representation and use, where the Lecture timetable will be represented in a matrix, and this matrix represents the chromosome. The length of the chromosome depends on the number of classrooms available, the time periods designated for study, and the number of days of the week as shown in the following steps:

1. Create a one-dimensional matrix, where its length is calculated as follows:

   Chromosome length =

   (Number of available classrooms) **X** (number of time periods (7 periods))

   X (number of study days in the week)
2. The number of lectures for each group is identified by specifying a number for each lecture as in *Table 11*.
3. The numbers that have been specified will be put in the chromosome randomly.

4. When putting the lecture numbers in the chromosome, this will result in a number of empty cells which is equal to its number:

   Number of empty cells = (chromosome length – number of lectures)

5. Must be put the **Zero** value in the empty cells to represent the lack of a lecture.

The Chromosome (table)

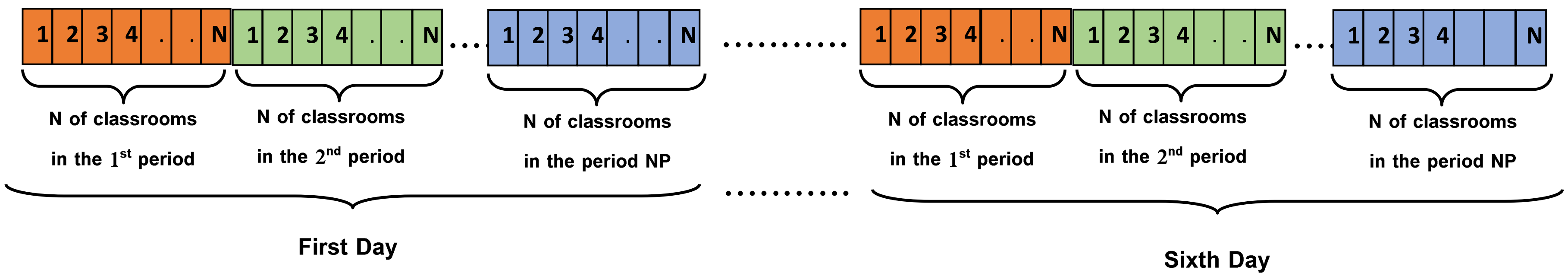


Figure 26: Representing the course timetable

Given to the importance of representation in genetic algorithms and to obtain satisfactory results, consideration must be given to the choice of the appropriate method of representation.

Accordingly, of the difficulty of finding suitable representation for such types of problems, which depends on the optimization, as well as; to the knowledge and evaluating the feasibility of using this representation in solving the problem and obtain satisfactory results.

The problem of timetables depends on several factors and the representation of all these factors; this requires a three-dimensional matrix and result in the difficulty in representation and dealing with this matter within the algorithms. Using the proposed representation in this thesis we will get less complexity and better representation and ease of expression in addition to the representation of the problem within a matrix with one dimension or a matrix with two dimensions and that according to the need during dealing programmatically.

Also, consider the symmetry between the problem and fitness function, so while ensuring that the representation to be implemented is appropriate and consistent with the procedures used in the genetic algorithms.

4.5.2 Representing the exam timetable

The digital representation is used to represent the exam timetable, where the exam timetable is represented by a matrix based on the number of available classrooms, the number of time periods reserved for examination, and the number of exam days as shown in the following steps:

1. Creating a one-dimensional matrix with a length calculated as follows:
2. Matrix length = (number of available classrooms) **X** (number of time periods (two periods)) **X** (number of exam days).
3. The number of courses is determined and a number is specified for each course.

4. The numbers that have been specified are put into the chromosome randomly.
5. When putting the courses numbers in the chromosome it will result in a number of empty cells that equal its number:

   Number of empty cells = (Matrix length – number of courses).
6. The value **Zero** must be put in the empty cells, which represent the lack of courses.

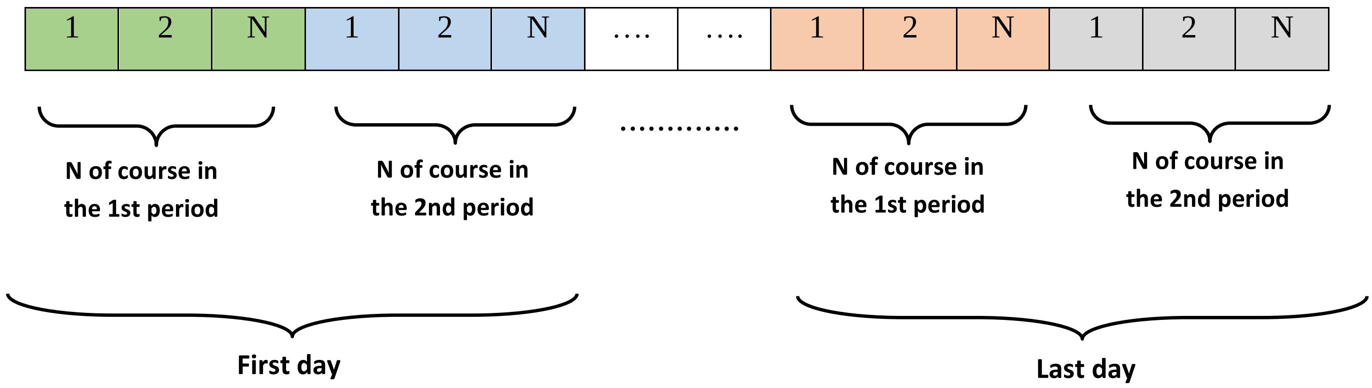


Figure 27: Representing the exam timetable

4.6 Assessing the quality of the solution

To assess the feasibility of the solutions and their importance in solving the problem according to hard and soft constraints, the problem will be represented in the fitness function to evaluate the solutions, and obtain solutions of high quality.

Because of many constraints and its difficulties in the timetables, we have transformed the timetable problem into finding the shortest path problem. To perform this transformation there are some steps that should be taken:

1. A five-column table should be made where the first column represents the lecture number; the second column represents the lecturer’s number; the third column the classroom number; the fourth column is the course study year, and the fifth column is the course number.

This table is represented by a matrix known as an **Array input matrix** for the easiness to deal with as is in the table below:

Table 14 : Input matrix

Lecture no.	Course no.	Lecturer no.	Classroom no.	Course year
1	1	45	8	1
2	1	45	8	1
3	2	22	11	1
4	2	22	5	1
5	3	45	6	1
6	3	45	7	1
10	5	100	9	2
11	6	57	35	2
30	15	12	12	3
31	15	34	20	3
50	25	3	1	5
51	25	3	7	5
.	....	....	....	.....
.	....	....	....	.....
.	....	....	....	.....
N	N	48	18	N

2. The exam timetable depends on the students' registration of courses during the beginning of the semester. This data should be recorded as described in the registration matrix in *Table 15*.

Table 15: Registration matrix

Lecture no.	Students Number							
1	123	145	11	888	790	44	.	N
2	286	45	100	8	237	1900	.	N
3	135	2342	1001	11	1975	31	.	N
4	2005	145	121	567	999	241	.	N
5	351	1213	286	237	2005	91	.	N
.	....	....	.....	....	....	.....	.	N
.	....	....	.....	....	....	.....	.	N
N	404	48	165	8765	....	.....	.	N

3. Using the graph theory formula G= {V, E}, where V represents heads and E, represents the ribs or edges.

After we represent the chromosome as shown in the figure below:

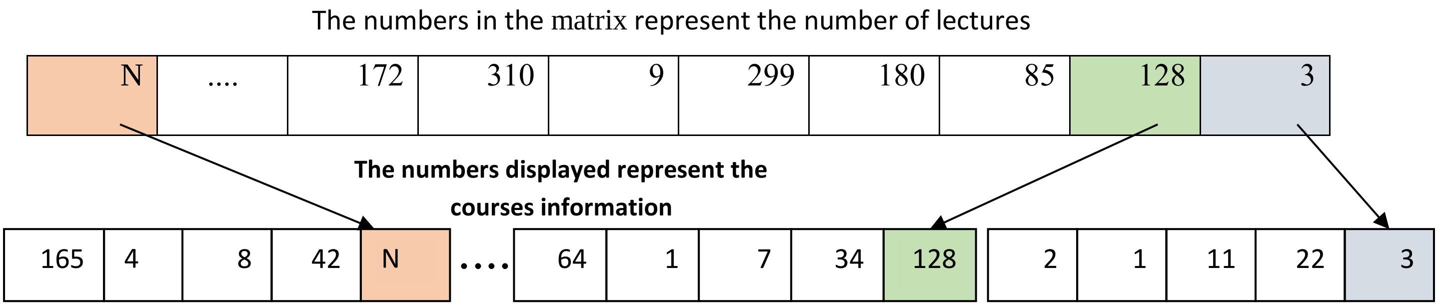


Figure 28: Represent the course timetable chromosome

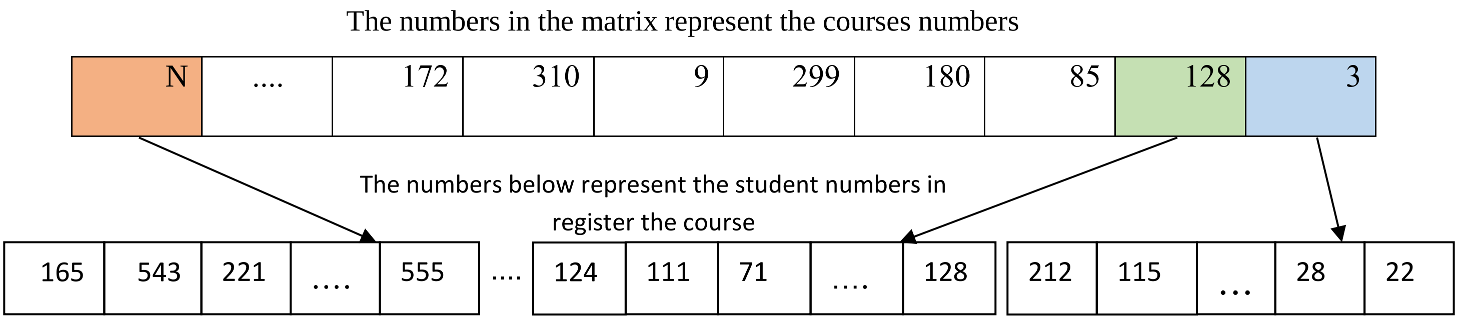


Figure 29: Represent the exam timetable chromosome

As shown in ***Table 14, 15*** through the Array Input matrix and Registration matrix, we will configure a matrix for weights, and through the matrix of weights, we can calculate the value of the fitness function. We will clarify the mechanism of dealing with the problem of finding the shortest path in line with the timetable problem, we consider that each chromosome from the chromosome represents a head in the graph and we need to calculate **E** between each two heads by the following:

If we consider the first head as (i) and the second head as (j) so the edge or rib ($\mathbf{E}_{i\,j}$) is calculated based on the following cases:

1. $\mathbf{E_{i\,j}} = 0$ IF i=j

   This case will result in the main diameter in the weights matrix with the value 0.

2. $\mathbf{E_{i\,j}} = 1$ IF (ArrayInput(i , 2) = ArrayInput(j , 2))

   The meaning of this requirement is that if there is a lecturer that gives more than one lecture in the same period in the same day, the number (**1**) should be placed in the first weights matrix (**WL**) and in case the requirement is not met, a (**0**) should be placed instead.

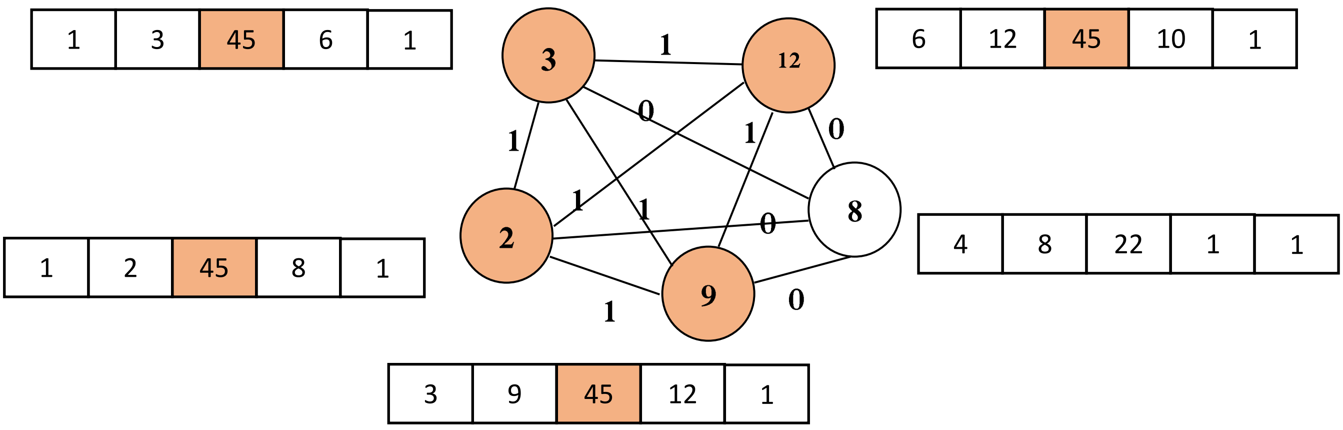


Figure 30: Graph for representation to clash a teacher

3. $\mathbf{E_{i\,j}} = 2$ IF (ArrayInput(i , 3) = ArrayInput(j , 3))

   This requirement means that is not allowed for two lectures to be held in the same classroom on the same day and time period. If the requirement is met the number (**2**) should be placed in the second weights matrix (**WR**) and if the requirement is not met then a (**0**) should be placed instead.

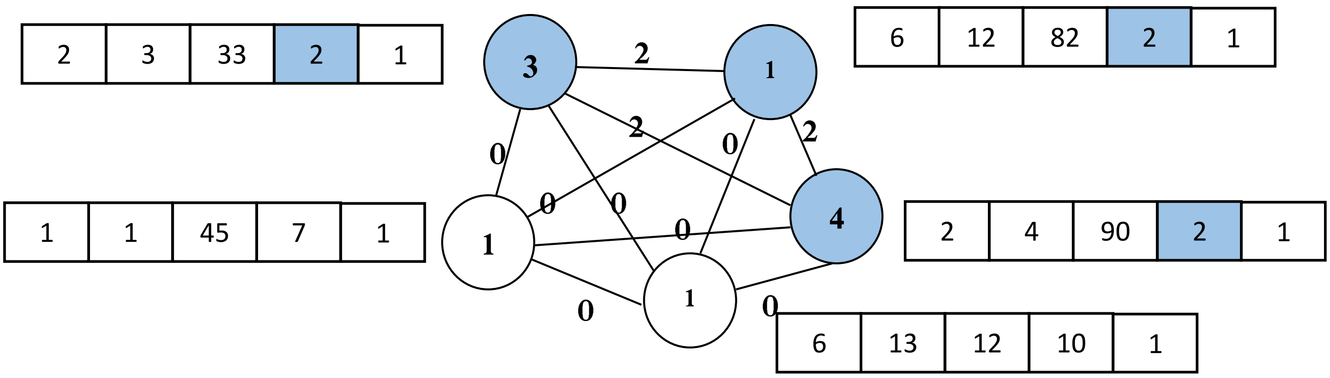


Figure 31: Graph for representation to clash in classroom

4. $E_{i\,j}$= 3 IF ((ArrayInput(i,0) = ArrayInput(j,0)) and (ArrayInput(i ,1) = ArrayInput(j , 1)))

This is the last requirement, where if there was a lecturer giving two lectures for the same course on the same day the number 3 must be placed in the third weights matrix (**WLC**), the goal of this test is to make sure that only one lecture will be given for the same course in one day. In the case the requirement isn't met, a (**0**) should be placed instead.

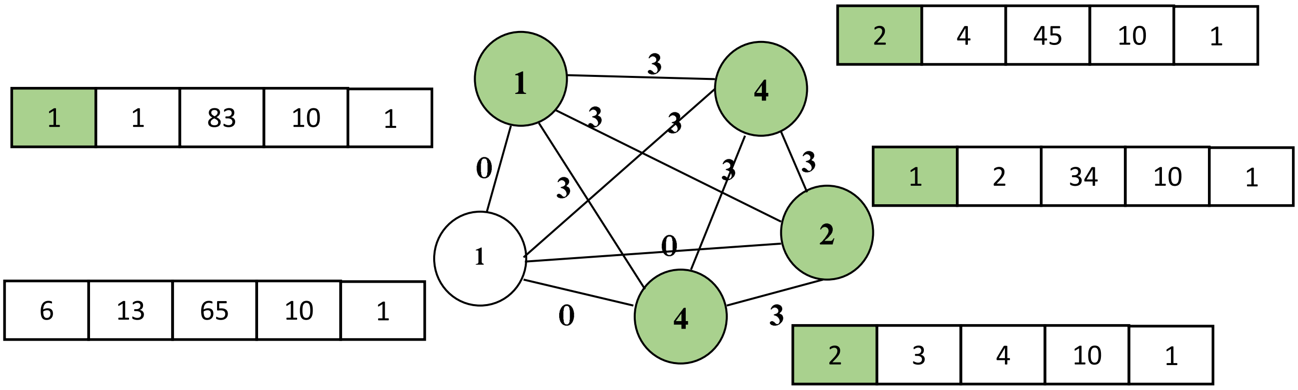


Figure 32: Graph for representation one lecture for the same course in the same day

5. $E_{i\,j}$= 1 IF (RegistrationArray(1, i) $\cap$ RegistrationArray(1, j) != 0)

In this case, the previous condition means that there may be the presence of two courses on the same day and the same period if there was at least one student only has registered in same courses during the semester. In the case of the condition should be placed one in the matrix of the weights (**WEC**) In the event that the condition must be (**0**).

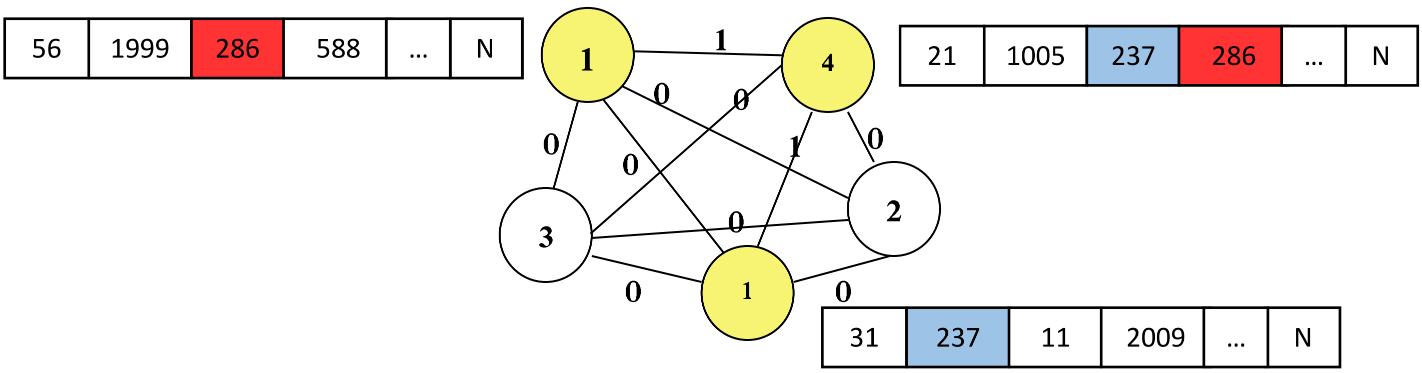


Figure 33: Graph for representation the conflicting course in the exam timetable

- ***Fitness function (quality solution)***

In order to measure the quality of the produced solutions, the fitness function is needed and indispensable. The quality value must be calculated so that we can make the appropriate decision in determining the feasibility of the solutions

The formulation that used to make sure the timetable solutions satisfied hard constraints, we could start creating the weights matrices, which is a two-dimensional matrix where the number of rows will start **(**from **1** to the **number of lectures**), and the number of columns will start **(**from **1** to the **number of lectures)**. The weights matrix will be filled according to the previous requirements and with the same method that has been displayed.

	1	2	3	4	5	6	N
1	0	1	0	0	1	1	1
2	1	0	0	0	1	0	1
3	0	0	0	1	0	0	0
4	0	0	1	0	0	0	0
5	1	0	0	0	0	1	0
6	0	1	0	0	1	0	1
N	1	0	0	0	1	0	0

Figure 34: The first weight matrix with values (Course Timetable)

	1	2	3	4	5	6	N
1	0	2	0	0	2	0	2
2	2	0	0	0	0	0	0
3	0	0	0	2	0	0	2
4	0	0	0	0	0	2	0
5	0	2	0	0	0	0	0
6	2	0	0	2	0	0	0
N	0	0	2	0	0	0	0

Figure 35: The second weight matrix with values (Course Timetable)

	1	2	3	4	5	6	N
1	0	3	0	0	3	0	0
2	3	0	0	0	0	0	3
3	0	0	0	3	0	0	3
4	0	0	3	0	0	0	0
5	0	3	0	0	0	3	0
6	0	0	0	0	3	0	0
N	3	0	0	0	0	0	0

Figure 36: The third weight matrix with values (Course Timetable)

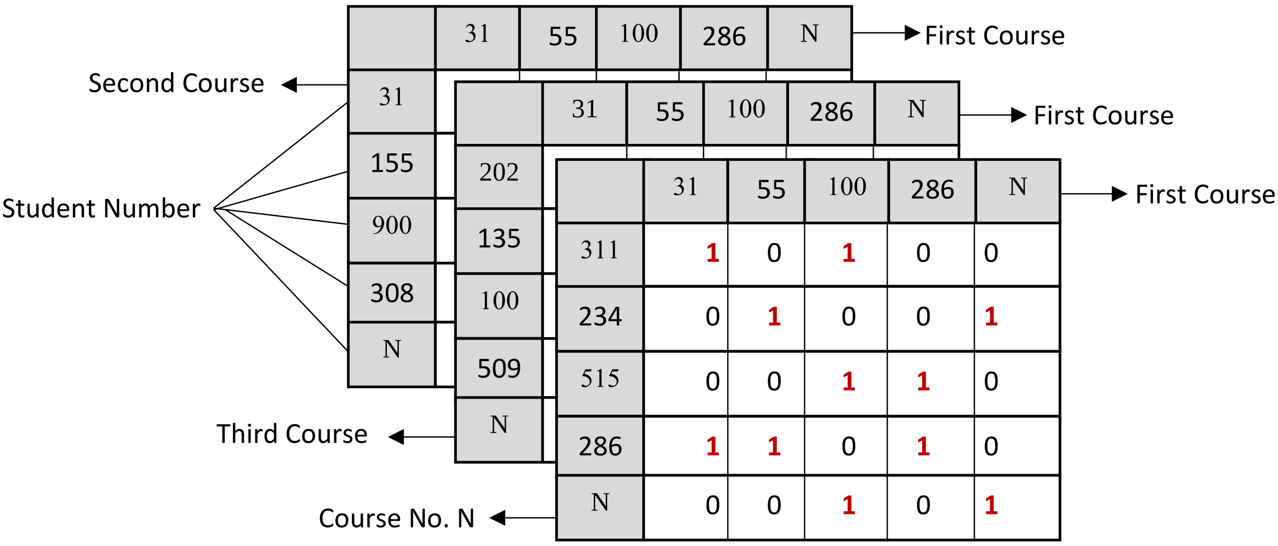


Figure 37: Weight matrix with values (Exam TimeTable)

- Through the weights matrices the fitness function can be calculated for each individual in the generation, where we will deal with first weights matrix (WL) and the second weights matrix (WR) and the third weights matrix (WLC) and that is by testing each chromosome in the chromosome and calculating the sum of each conflict in the chromosome as shown below:

FT = $\sum_{i,j=1}^{n}$ **WL (C(i) , C(j))**

Where **C** represents the matrix that contains the chromosome and **WL** represent the weights matrix that maintains the conflicts of the teaching faculty.

$\mathbf{FC} = \sum_{i,j=1}^{n} \mathbf{WR}\ (\mathbf{C(i)}, \mathbf{C(j)})$

Where **WR** represents the weights matrix that maintains the classroom conflicts.

$\mathbf{FD} = \sum_{i,j=1}^{n} \mathbf{WLC}\ (\mathbf{C(i)}, \mathbf{C(j)})$

Where **WLC** represent the weights matrix that maintains in not teaching more than one lecture for the same course in the same day.

$\mathbf{FEC} = \sum_{i,j=1}^{n} \mathbf{WEC}\ (\mathbf{C(i)}, \mathbf{C(j)})$

Where **C** represents the matrix that contains the exam chromosome and **WEC** represent the weights matrix conflict for students in the final exam for the same time in the same day.

Quality of solution LTT = FT + FC + FD

Quality of solution ETT = FEC

The value of the **fitness function** represents the number of conflicts in the timetable. The role of the gene algorithm to decrease the number of conflicts or improve the quality of solutions until we reach a solution with the quality of **zero**, which means a timetable with no conflicts.

4.7 The Gene Algorithm used for solving the timetable problem

- Create two timetables randomly because the crossover technique It will produce more than 100 individuals in every generation through parents and does not take into consideration the restrictions or conditions in preparing the tables. Which means that all generations even the initial generation does not consider the restrictions and conditions, but creates tables randomly instead.
- Should test the quality of the solution. Where the fitness function tests the timetable and identifies the conflicts in the tables and gives values that represent the number of conflicts in the table.

- Implementation crossover chromosomes Among the members of the generation using multi-crossover techniques that have been implemented and executed on this problem in this research.

- The mutation process must be executed as well. The changeable genes were chosen, where the probability of a mutation on three genes are interchangeable on the same chromosome.

- Implementation the fitness function again because a new generation containing a group of individuals (the group of solutions) has been produced. Therefore, the quality of each solution should be tested. Where the number of conflicts in each solution would be identified, which means the result of this process is numbers that represent the quality of a solution. for example, the number **zero** represents a timetable with no conflicts and the number **ten** represents a timetable with ten conflicts and so on.

- After all the previous stages are done, a timetable free of conflicts should be chosen. Which means a timetable with a **zero** value quality of solution should be chosen. It is also possible to obtain more than one table free of conflicts.

The Algorithm

1. ***Start***
2. ***Enter the number of classrooms NR and the number of lectures NL and the number of days ND and the number of time periods NP***
3. ***Primary generation formation***

/* the primary generation is formed from two individuals only (two timetables) and that is because of the crossover technique used.

4. ***Calculate the quality for each individual in the generation***
5. ***If quality bigger than zero: YES, go to step 7***
6. ***Store the individual with zero value quality in the matrix, then go to step 9***
7. ***Call the crossover process***
8. ***Call mutation process, then go to step 4***
9. ***End***

Chapter 5

The Proposed application and the experimental results

5.1 Introduction

In the previous chapters, we have discussed the details of the application design, which related to the timetable problem and the methods used to address this problem. Also, we explained the basic concepts of the timetable process for (lectures and examinations) and all possible restrictions. Moreover, we have presented the concepts of the genetic algorithms and the general algorithms used. Eventually, we presented a proposed method for solving the timetable preparation problem using genetic algorithms.

In this chapter, a case study will be discussing. The case study presents details preparation of a timetable for the Faculty of Information Technology, Benghazi University – Libya. We will clarify all the requirements and criteria to identify the table quality, as well as; identify the timetable preparation requirements.

The proposed algorithm that is described in the previous chapter, we will describe the implementation techniques in this chapter, and show tested results and comparing the results with the requirements and criteria identified in Chapter 2; Section 2.3.

The goal of this thesis is to automate the process of timetable preparation according to the official nature of our case study about the faculty of Information Technology. The designed system will work on preparing and generating timetables based on the suggested algorithm by using the genetic algorithms.

5.2 The application Design

Construction of the proposed system in this thesis depends on three basic phases (Input Data, Applied and output phase).

In the first phase must enter all data, information and requirements preparation timetable (days, slots, staff name, rooms name, etc.) considered as the most important phase So that we can complete the rest of the phases Successfully.

Moving to the second phase after making sure to enter all the timetable preparation requirements the applied phase comes to executes smart genetic algorithms proposed to generate the many of timetables using MATLAB, then stores the resulted timetables in the Excel file.

Finally, the last phase is the output phase contains an integrated system for management and control of the data where system it provides user interfaces to use the system easily also provide decoding Process after the completion of the applied phase that generated the timetables. Accordingly, can use print function to print many of reports exam, lecture, room and staff timetables and other reports as shown in Chapter 4. Figure 38 explains the initial design of the system with its full contents.

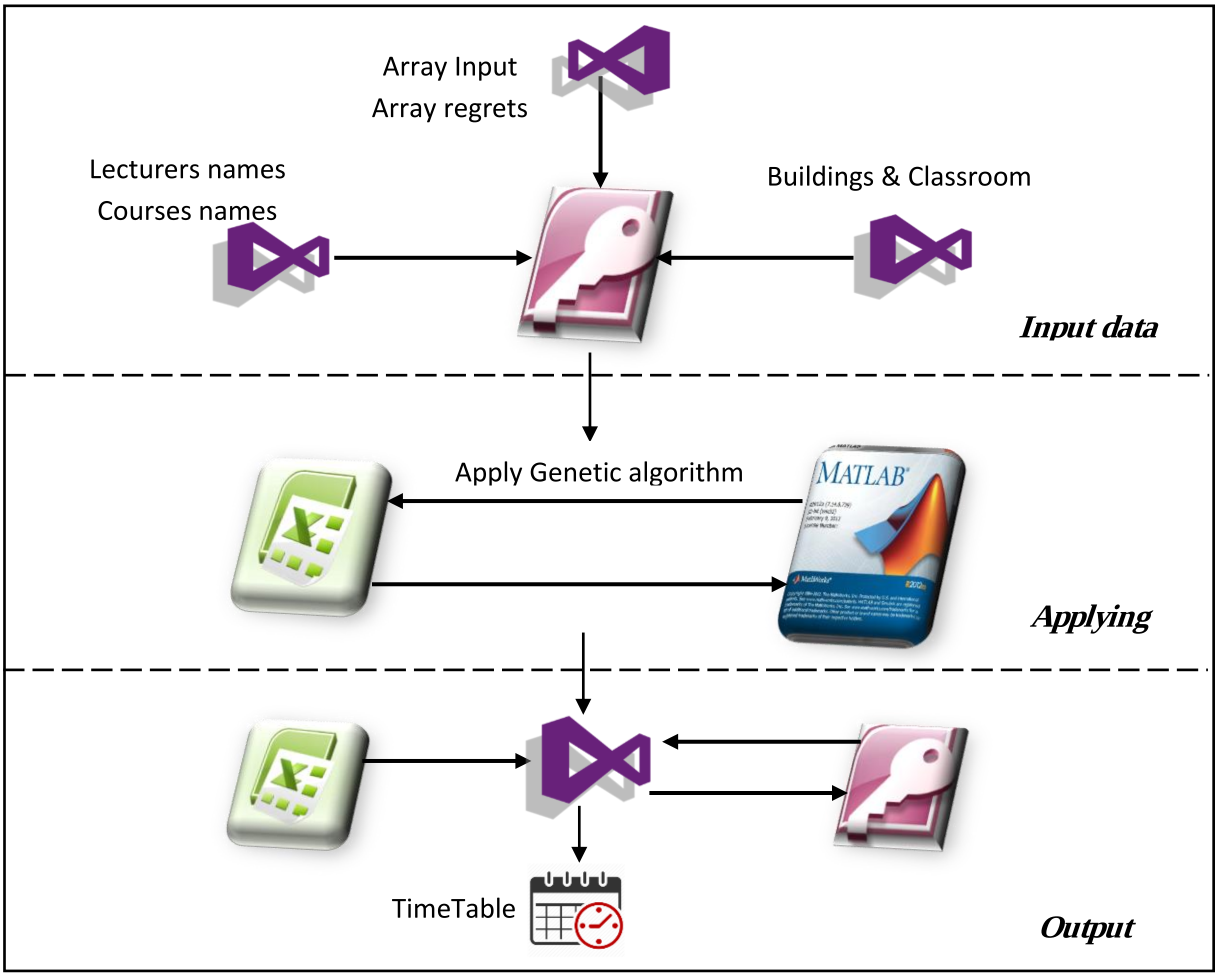


Figure 38: The general design of system

The reason for choosing MATLAB is the ease of dealing with matrices, also; the ease of configuring and aligning matrices according to our desire. As a result, of executing the procedures mentioned earlier, MATLAB will produce a matrix containing the lecture numbers. Also, this matrix does not have any conflicts, where all restrictions and conditions apply to it. This means that the course timetable has been obtained, but there is a problem! This matrix is only numbers, thus; it means we cannot deal with or understand these numbers. Therefore, there is a need for a system that decodes these numbers, where the system takes each number in the matrix and represents it with the relevant data. The decoding process should be done for each number until all the numbers in the matrix are completed.

The Visual Studio Application Development has been used to decode and fetch the data, then place it in a report (course timetable) that can be printed.

To begin the work, it was necessary to use real data to test the work of the genetic algorithms and the quality of the solutions resulting from the algorithm, therefore; fact data has been used for the technical college of information- Benghazi University.

5.3 The application input

It is important to know and identify the valid inputs that needed to be used in the genetic algorithm, and all these inputs should be entered in the Visual Studio program to be dealt with them later using MATLAB. The knowledge of the inputs is not enough to start the algorithm's work, we must start with clarifying these inputs even we can deal with them using genetic algorithms. Firstly, we will identify the system inputs in the Visual Basic program as follows:

1. *The names of faculty members for the semester should be identified and inputted using Visual Basic.*
2. *A list of available halls, classrooms and its capacities should be identified.*
3. *A list of the time periods for each day should be identified.*
4. *A list of all the lectures for the semester should be identified.*
5. *A list of all the course names that will have final exams should be identified*
6. *Enrollment documents should be extracted for each student within the courses for the semester.*

According to our case study in the Faculty of Information Technology – Benghazi University, the inputs are as follows:

Course timetable inputs:

- Number of classrooms: 16 classrooms
- Number of days: 6 days
- Number of time periods: 5 time periods
- Number of lectures: 330 lectures

Examination timetable inputs:

- Number of courses: 88 course
- Number of students in each course based on the semester enrollment
- Number of days: 12 days
- Number of periods: 2 periods
- Number of available classrooms: 20 classrooms

5.4 Implement the proposed design using genetic algorithms

In Chapter 4; section 4.5, we have explained how to represent a course timetable and the algorithm used. However, we have now to explain the method of implementing the representation which has been put forward previously to obtain the chromosome. Consequently, since there is a series of numbers and each number represents a certain lecture, we have now to start explaining how to implement the genetic algorithm.

In the beginning, a one-dimensional matrix should be created with length equal to:

Length = number of days ✖ Number of periods ✖ Number of classrooms

This matrix is used to accommodate the courses during the implementation phase.

Should be filled-in the matrix with values (random numbers from 1 to the matrix length), these values represent lectures, and then a function that generates random numbers will be used to generate random numbers from one to the number inputted.

For the examination timetable, in accordance with, identifying the targeted courses which will be included in the examination timetable, a matrix is created using the course numbers that obtained from the student enrollment documents in the beginning of the semester, where these numbers are stored in the courses matrix.

Two matrices should be created using this method considering them as the primary generation according to the technology used in transit as explained previously in Chapter 2; Section 2.4.4.5.

By doing so, we have completed the first stage which is generating a primary generation for the genetic algorithm, the following Figure 39 and Figure 40 display a chromosome sample.

422	**66**	**102**	**10**	**322**	**43**	**122**	**55**	**9**	**54**	**65**	**.**	**L**

Figure 39: A chromosome in a course timetable

722	**666**	**702**	**710**	**622**	**643**	**672**	**655**	**709**	**654**	**665**	**.**	**NE**
55	**14**	**33**	**35**	**1**	**2**	**11**	**7**	**34**	**3**	**9**	**.**	**NE**

Figure 40: A chromosome in an examination timetable

Algorithmic processes will be performed on this generation which is the chromosomal passage and mutation. Finally, the choice is done by using the quality function, where it measures the quality of each chromosome or individual in the generation.

This matrix is considered the representation of the course timetable and considered a chromosome (solution). We also have to start with two tables, this is meaning the primary generation contains two chromosomes so that the chromosomal transit process can happen between them.

5.5 The Results

5.5.1 Course TimeTable Results

Based on the results obtained from applying the genetic algorithms that are proposed in Alaraibi E. (2012). The work done in this thesis is to improve the previous work and achieve higher efficiency and accuracy in less time, moreover, an exam timetable is presented at the same time to synchronise the two-time tables, and to help the staff working in academic departments to take better decisions.

Work has been done to improve the process of establishing the initial generation using the multi-crossover technique, resulting in a larger number of individuals in one generation in less time than in the previous algorithm referred to earlier. In addition to the development of the fitness function in the proposed genetic algorithm in this thesis using the graph theory by giving the weights between the nodes, which represents the value of the conflict between courses, which saved less consumption of resources and sources and obtain better results in less time as shown in the *Tables 9, 10*.

In this part of the thesis, the results obtained will be presented and compared to the previous work related to the same technique used to solve timetable problem, to determine the importance and feasibility of the proposed genetic algorithm in the thesis. We'll explain the procedures for reaching results and display the results obtained from the implementation and experiments.

Near finishing the representation and the input process, will obtain a group of matrices. Each matrix containing a group of numbers, where each number represents a group of data. Thus, we are in one step away from implementing the algorithm procedures that have been identified earlier, and by implementing these procedures we will obtain a matrix containing a group of numbers that are then stored in an excel file (xls). Completed the file; therefore, must be then transformed into an access file (accdb). The goal of transforming the file from Excel to Access is to use the data by the developed application by Visual Basic, which is used to express the results in many forms.

Through conducting some experiments on the process of generating timetables, the results proved that all the results are free of any conflicts, and all constraints that identified are applied correctly.

The following *Table 16* and *Table 17* have displayed some samples of the generated results.

Table 16: Result without using Graph theory in Fitness

Exp. No.	Type TimeTable	Fitness	Time	Iteration
1	Course TimeTable	120-140	21.3033	6538
2	Course TimeTable	120-140	15.982	4507
3	Course TimeTable	120-140	11.9086	3681
4	Course TimeTable	120-140	14.1469	4103
5	Course TimeTable	120-140	18.521	5008
6	Course TimeTable	120-140	10.0010	3403
1	Exams TimeTable	18-20	32.9865	9986
2	Exams TimeTable	40-45	37.339	15299
3	Exams TimeTable	30-35	29.8647	8988
4	Exams TimeTable	45-50	40.293	17871
5	Exams TimeTable	20-25	27.9891	8769
6	Exams TimeTable	35-40	44.162	19753

Table 17: Result with using Graph theory in Fitness

Exp. No.	Type Time Table	Fitness	Time	Iteration
1	Course TimeTable	120-140	1.8771	716
2	Course TimeTable	120-140	2.5281	1100
3	Course TimeTable	120-140	2.7008	1189
4	Course TimeTable	120-140	1.7904	765
5	Course TimeTable	120-140	2.0219	865
6	Course TimeTable	120-140	2.3979	1045
1	Exams TimeTable	25-30	7.5865	11736
2	Exams TimeTable	20-25	17.2354	17620
3	Exams TimeTable	30-35	6.7979	10929
4	Exams TimeTable	20-25	3.773	6543
5	Exams TimeTable	25-30	6.468	10007
6	Exams TimeTable	30-35	7.901	11950

The previous values in *Table 16 and Table 17* illustrate the difference in the importance of using the graphics theory and its ability to improve the time of obtaining results within predefined constraints.

Consider the values and comparing the time of production of the schedules, the large difference in the speed of obtaining a schedule (course and exams) without occurs any conflicts in results generated. In addition, ability to producing more individuals in one generation by shorter time in the case of use fitness function is without the graph theory. The differences production time of the timetable shows in Figure 41 and Figure 42.

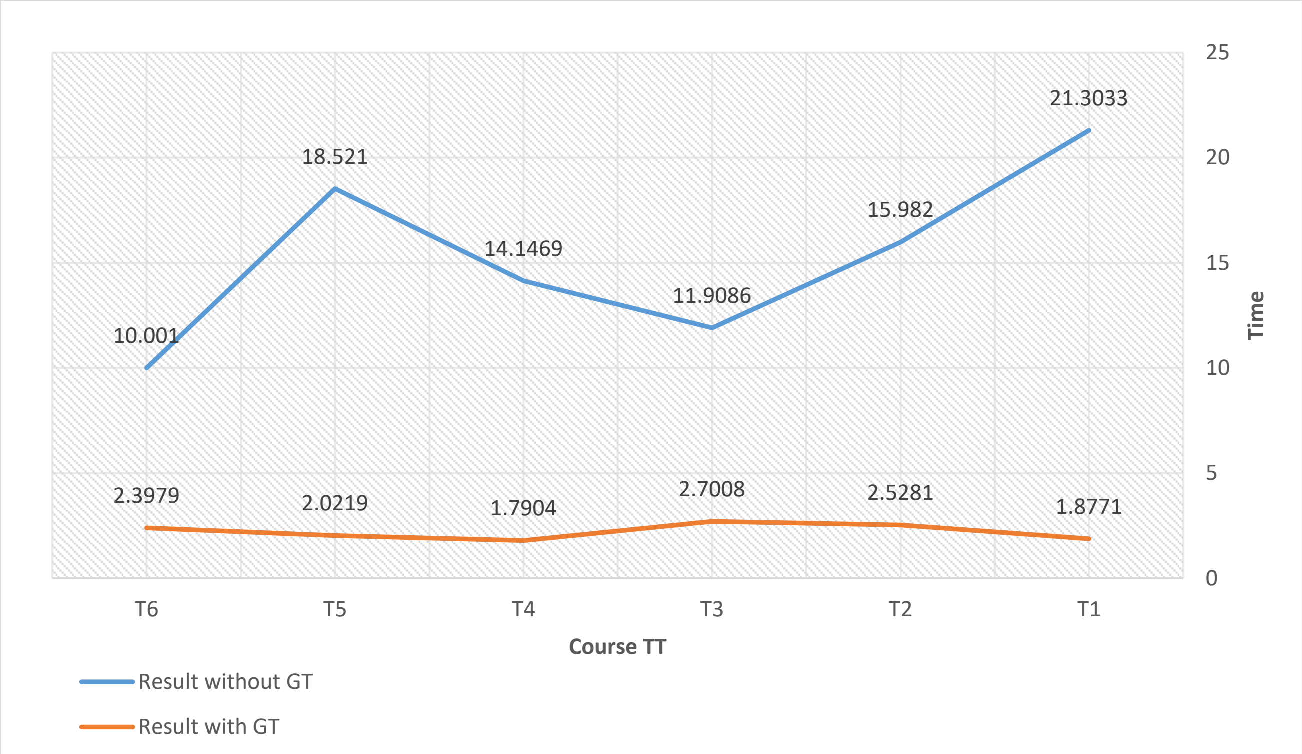


Figure 41: Curves show the times for produce course timetable

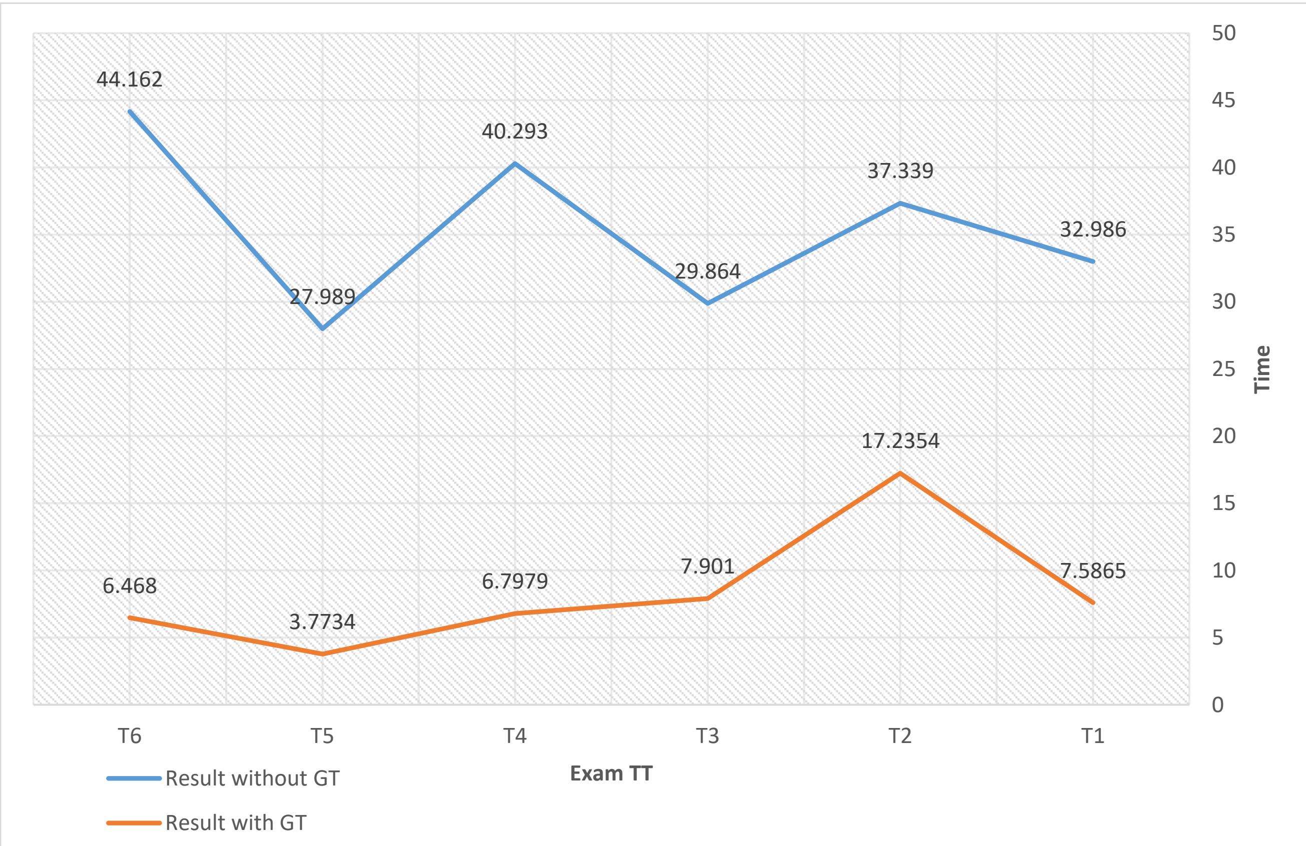


Figure 42: Curves show the times for produce exam timetable

The fitness function to reach the results in a less time, where the time depends on the acquisition of a table according to the individuals that were generated randomly. In addition, the time is different according to these generations. Figure 43 shows a sample of the course timetable that was produced through the system for the generation of timetables.

By conducting some experiments, the system was operated for 12-hours. More than 120 schedules were produced during the previous period, and it was verified manually and programmatically that each of these tables is completely different from the other.

CTT

id	idd	f1	f2	f3	f4	f5	f6	f7
1	1	42	0	0	43	200	193	0
2	1	23	69	19	119	12	0	126
3	1	0	2	0	71	280	56	97
4	1	66	267	67	164	145	0	0
5	1	310	269	287	0	14	107	243
6	1	222	116	0	0	73	0	322
7	1	0	161	202	49	272	141	266
8	1	131	0	26	0	241	246	323
9	1	54	4	210	0	0	0	206
10	1	305	0	148	0	0	0	0
11	1	45	0	229	112	0	224	0
12	1	311	174	0	281	0	0	0
13	1	187	84	252	178	0	133	239
14	1	102	153	328	297	0	0	0
15	1	290	105	121	181	185	60	254
16	1	227	0	0	276	0	9	0
17	2	89	0	78	143	0	16	0
18	2	0	0	236	0	329	199	249
19	2	0	44	0	0	0	0	0
20	2	0	0	271	72	0	295	0
21	2	0	321	152	0	230	0	91
22	2	41	0	24	204	100	98	17
23	2	33	53	273	0	11	0	0
24	2	0	0	0	195	0	316	47

Record: 1 of 96 No Filter Search

Figure 43: A Sample of a course timetable

Figure 43 showed a sample from the first-course timetable generated using MATLAB which is stored in an excel file (xls), and the time required to generate this table using the genetic algorithm is **1.9771** minutes. Figure 44 displays the table's curve:

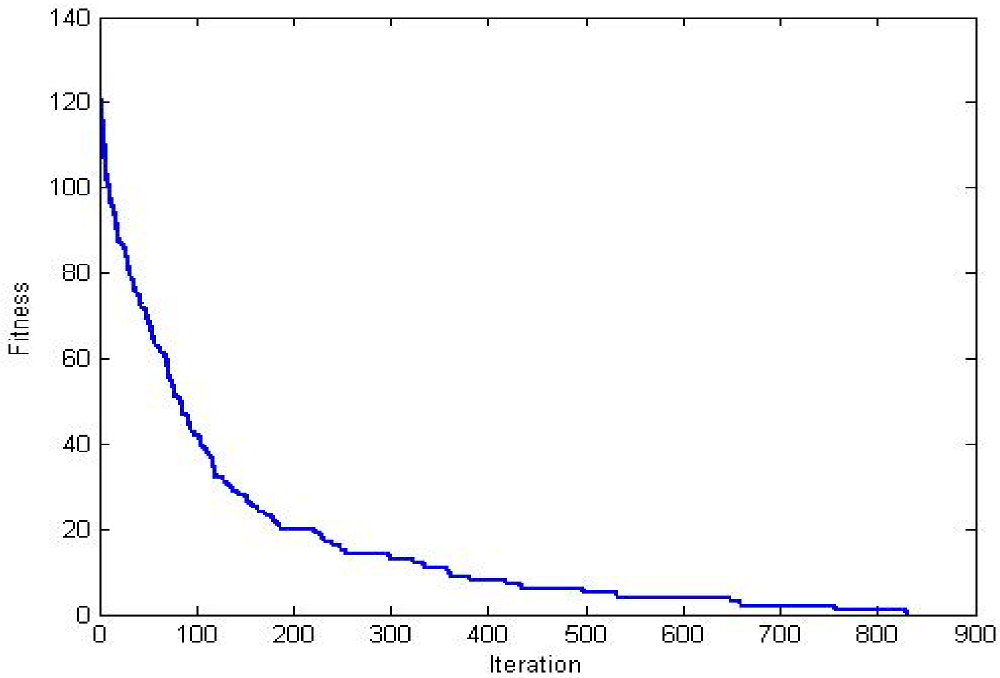


Figure 44: A curved production course timetable

5.5.2 Exams TimeTable Results

With respect to the examination timetable, it took more time to produce the timetable because the probability of the student's conflict was greater, also; the data was limited by two time periods and the duration of the exams was only two weeks. Thus, producing of examination timetable is more complex than the course timetable.

By conducting some experiments, the system was run for 12 hours, where more than 65 timetables were generated during this time. It was confirmed programmatically and manually that each of these tables is completely different from the other.

ETT

id	idd	f1	f2
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0
7	1	53	38
8	1	0	48
9	1	40	0
10	1	0	55
11	1	0	0
12	1	0	0
13	1	0	0
14	1	0	26
15	1	0	0
16	1	0	76
17	1	0	0
18	1	0	0
19	1	0	0
20	1	0	0
21	1	0	0
22	1	0	0
23	1	0	0
24	1	0	0

Record: 1 of 288 No Filter Search

Figure 45: A Sample of an examination timetable

Figure 45 shows a sample from the first examination timetable that was generated using MATLAB, where it is stored in an excel file (xls) and the time required to generate this table using the genetic algorithm is 3. 7734 minutes. The Figure 46 displays the table's curve:

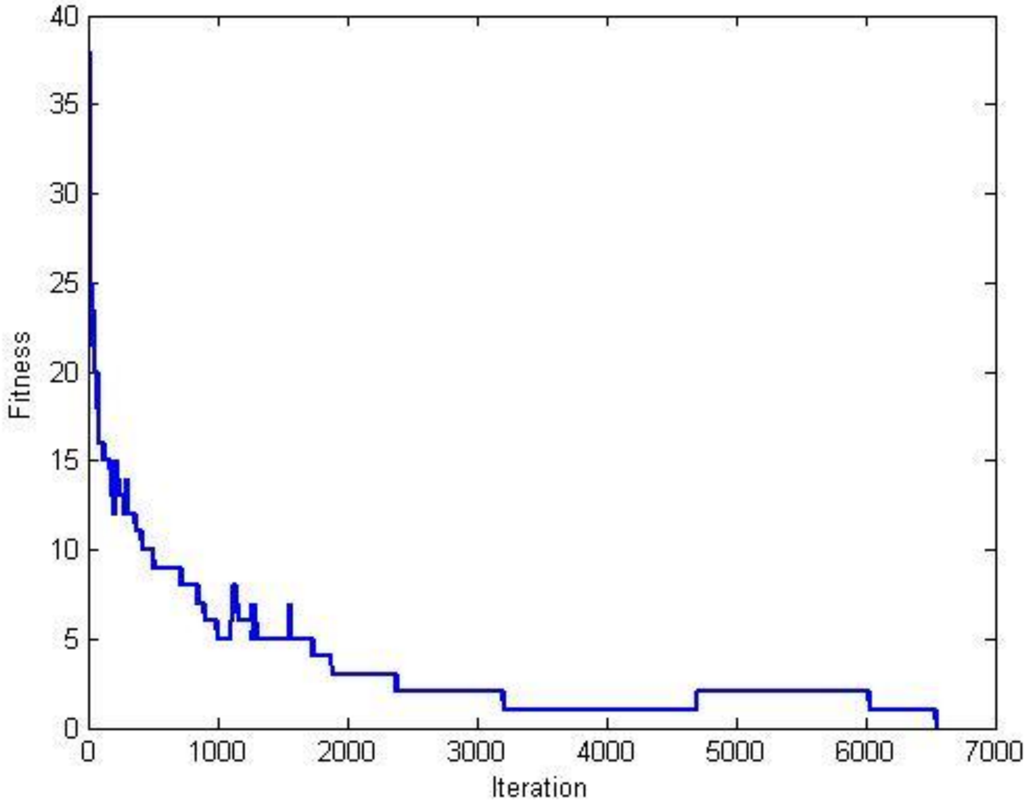


Figure 46: A Curved production examination timetable

Finally, in this chapter, we concluded that the results obtained are precise according to the accuracy of the data that are entered into the data management system which is responsible for their validity. In addition, these results are not fixed; they are variable according to the inputs by users of the system. Moreover, each implementation process produces a new schedule which is different from the table that has been produced even if the data has not changed from the user.

5.6 Comparison of results

5.6.1 Course TimeTable

Based on the above results, and the results obtained from literature and previous studies described in the chapter 3, we will compare our results with the previous studies results as following:

Through the narration of literature results, we compare our results with the literature results as in *Table 18*.

Table 18: Comparison of course timetable results

Literature	Iterations	Fitness Value	Best Fitness	Time (secs)	No. course	No. rooms	Periods
Min Wang(2012)	100	630 ± 10	--	520.9020	40	5	20
Teoh Chong keat(2012)	100	0	0	592.9	50	--	--
Walid Y. (2011)	1000	15617	20000	2880	--	--	--
CTT(our results)	765	0	0	107.424	130	16	7

The comparison of these studies in *Table 18* shows that the time for producing a lecture timetable is free of conflicts in this study and was the best results, where if the number of attempts is compared, the research is the least number of attempts to obtain a conflict-free timetable in the least time between these times.

5.6.2 Exams TimeTable

Finally, the comparison between our results and literature was presented in the examination timetable problem presented in chapter 3. *Table 19* shows the results.

Table 19: Comparison of exam timetable results

Literature	Iterations	Fitness Value	Time (secs)
Twinkle Dahiya et.al (2015)	180	26.1058	--
Azman Yasin et.al(2007)	100	2770.0	406
ETT(our results)	6543	0	226

The exam schedule depends on resources and constraints that are completely different from the lecture schedule. If the results obtained are compared with the results of the previous studies, the results were also better in terms of the number of generations' production attempts, which represent the examination schedules compared to the schedule generation time. More than 6000 attempts to generate a generation were reached to an individual free of conflicts in one generation and did not take more than 3 minutes compared to the times of generation of larger tables compared to the number of attempts to generate generations less than what was presented in the research.

Finally, the differences in the data and the analysis methods in the literature may not presents which study better than the other, but may evaluate the importance and the feasibility of this work by finding a number of possible solutions for the timetable problem, which is meet all the difficult constraints and also takes into account the greatest amount of soft constraints to improve the quality of these solutions in the shortest possible time.

It is also possible to evaluate the importance of this research on what has been presented in the previous studies about the speed of finding the best solution from solutions that can be obtained in a time is negligible compared to the previous studies. In addition, the genetic algorithm presented in this study had the ability to produce the largest number of individuals in the generation at a time less than previous studies. This adds greater ability to produce more generations in less time because of the use of multiple chromosome crossing technologies.

With regard to the ability of algorithms to produce large numbers of schedules according to the desire of users of the system in determining the number of required to generate schedules (study and examinations).

The reason for the high capacity is the rapid assessment and calculation of the quality of individuals within the generation and the choice of the individual to whom the restrictions apply. The reason for this is the use of the theory of schematics in the quality function, which adds a higher ability to calculate the speed of the quality values of individuals that are generated in greater numbers.

From the above we can be summarized the strength of this thesis in the follows:

- The increase in generation of individuals in the generation has become less time-consuming.
- It is assumed that increasing the number of individuals results in a long time in calculating the quality of these individuals within the generation. This problem was overcome by using graphics theory to calculate the quality of individuals faster, resulting in better performance of the genetic algorithm.

Chapter 6

Conclusions and Future Work

6.1 Summary

Timetabling is one of the common challenging scheduling problems, in which we need to maximise the allocation of resources and minimise the violation of constraints. Timetabling problems are often made complicated by the details of a particular timetabling task and are often considered to be NP-hard problems. The university timetabling problem (UTP) is one type of timetabling problems. This problem has attracted significant research interest over the past decades and is considered to be very difficult to solve due to its highly-constrained nature and the exponential growth of the size of the search space with the problem size.

The efficiency of any solution is achieved through the quality of the solution, the time spent to develop the solution, its feasibility, and the extent to which the requirements and limitations are met in the resulting solutions. The good measurement of an algorithm is achieving quality results in the least amount of time, which it has been achieved in our proposed algorithm to solve the timetable problem.

6.2 Conclusions

In this thesis, a new genetic algorithm chromosome representation plus new crossover technique has proposed by using graph theory in fitness function to solve a timetabling problem. In this thesis, we studied the traditional genetic algorithms for the simple UTP, and then we come up with new ideas to enhance the performance of traditional genetic algorithms for solving the UTP with the challenge of Post Enrolment Course Timetabling Problem. Hence, we described the complicated nature of university course timetabling.

In addition, the new techniques have been used the GA such as Multi-crossover technique, which was explained in chapter 2; Section 2.4.5 where a primary generation of only two individuals is created. By using this technique, the implementation time has been reduced and the achieved results were better than the results achieved in the previous studies in chapter 3.

The tool implemented using Matlab language and Visual Studio. The Case study uses an actual data derived from the Faculty of Information Technology, University of Benghazi. Were the results reached are met all hard and soft basic constraints to prepare timetables explained in Chapter 4. The average generation course timetable was two minutes however average generation of the exam timetable was five minutes. Change in timetables data (for example increase or decrease in the number of lectures, number of room, number of periods, etc.) affects the timetables generation time which depends on two important criteria which are more resources, less time you get the timetable, and fewer resources available, the longer time to get the timetables.

Finally, the results we achieved were successful. This work can be applied to any scheduling problem and is thus valuable contribution both to the academic field and practice. The results of this research have enabled the design process and timetable preparation to be transformed from a manual procedure to an automated procedure with minimum effort.

Reinforcement of the timetable can be established by adding more relevant restrictions before the evaluation process specified for the quality of the solution. In addition, the system output accuracy depends on ensuring updated data input and reflects the truth, as well as; it is free of defects and impurities.

Moreover, through this work, the proposed genetic algorithms with the data management system produced course schedule and examinations schedule with an estimated time of average 5 minutes per schedule. As well as, the possibility of producing an unlimited number of schedules during different periods of work, taking into account the difference between the tables that are produced and generated.

Also, the proposed solution works on the completion of the job once a timetable that meets all the subjected restrictions, whether difficult or flexible and the results obtained are shown in *Table 20*.

Table 20: Samples time spent for produce timetables

No.	Time spent	Type	Time spent	Type
1	1.8641	Lecture TimeTable	4.5359	Exam TimeTable
2	2.8635		7.2648	
3	3.0055		8.3286	

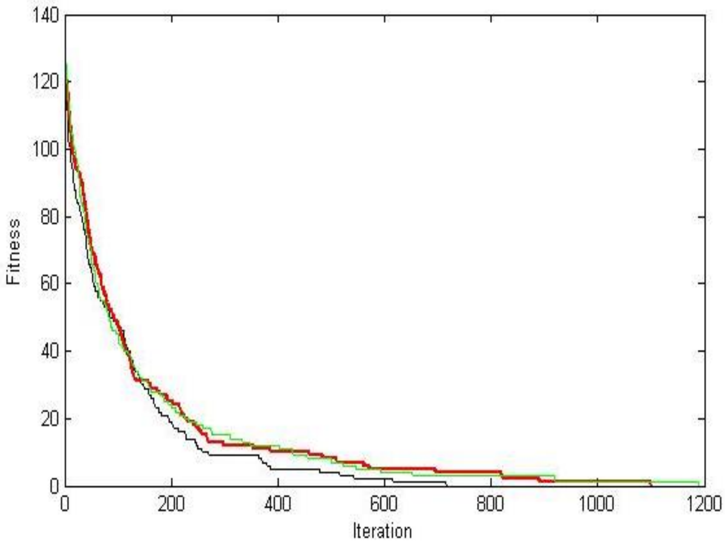


Figure 47 A curved production three-course timetable

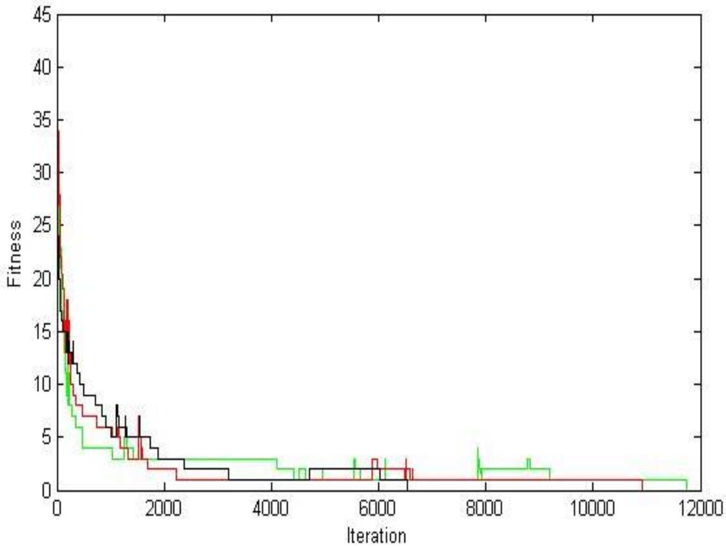


Figure 48 A curved production three-Exam timetable

Conducting many random experiments and using the same empirical data and real data, where we cooperated with the Faculty of Information Technology-University of Benghazi to use this data in the experiments, which is clear to us that these results after being reviewed and verified are free of conflicts and conform to all the constraints previously identified In chapter 2; Section 2.3.1 as in Figure 47 and in Figure 48.

In conclusion, the thesis contributions are as follows:

1. *Develop an intelligent genetic algorithm by using new GA techniques which used to generate solutions and best choices.*
2. *Develop a system for data management and reporting.*
3. *Ease of generating study timetable that sensitive to the subjected restrictions for the case which has been studied.*
4. *Ease of generating an examination table based on student lecture registration without conflict in examinations for students.*

Moreover, through the obtained results, the benefits of this research lies in:

1. *Capacity management of the resources available from the halls and using it appropriately without the waste of time or space.*
2. *The ability to optimize the allocation of lectures for professors.*
3. *The ease in providing reports for all timetable resources in record time without effort.*

6.3 System limitations

In this research, there a number of restrictions that can be seen in the future, that includes:

1. The length of a lecture cannot be separated into separate lengths. (For example, 3 hours lectures cannot be separated into 2 hours and 1-hour lectures).
2. Lectures that demand the use of computers (Labs) cannot be taken into account.
3. The Hard and soft constraints are constantly based on the case that has been studied and is not editable.

6.4 Recommendations for future work

Although the results from this research are more efficient from many previous works, the research can be strengthened in the future through developing a data management system that allows users to construct restrictions timetable to introduce them to the system, in addition; generate study timetables according to the restrictions that have been inputted, and which can be later translated into timetables to achieve better solutions.

Furthermore, the intelligent system that was implemented allows the user to prepare timetables to control the process of dividing lectures during the academic week and the importance of allocating lectures in the correct times and places. For example, customizing lecture requires a laboratory to be a lecture in the laboratory rather than being a hall to ensure the perfect use of available resources even as not to waste the resources.

Moreover, additional work has to be carried out to improve the performance of the algorithm by adding other genetic techniques, to become a smart and hybrid to produce results in less time than the acquired time.

References

Anderson A and Simpson AR. (1996). “*Genetic Algorithm Optimisation Software in FORTRAN”,* Research Report No. R136. Department of Civil and Environmental Engineering, The University of Adelaide.

Anshuman Sahu, Rudrajit Tapadar (2007). “*Solving the Assignment problem using Genetic Algorithm and Simulated Annealing”*. AENG International Journal of Applied Mathematics, 36:1, IJAM_36_1_7.

A. Schearf (1999). “*A survey of automated timetabling*". Artificial Intelligence Review,13(2): 87–127.

A. Wren (1995). “*Scheduling, timetabling and rostering – a special relationship?*". Proceedings of the 1st International Conference on the Practice and Theory of Automated Timetabling (PATAT I), Lecture Notes in Computer Science, vol. 1153, pp. 46–75.

Azman Yasin, Ghaith M Jaradat, Ruslizam Daud (2007). “*Secondary School Examination Timetabling using Genetic Algorithm*". Malaysia-Japan International Symposium on Advanced Technology 2007, MJISAT 2007, 12th -15th November, Kuala Lumpur, Malaysia.

Babaei, H., Karimpour, J., Hadidi, A. (2015). “*A survey of approaches for university course timetabling problem*". Computers and Industrial Engineering 86 pp. 43 - 59. DOI:10.1016/j.cie.2014.11.010.

Banzhaf, W. (1990). “*The Molecular Traveling Salesman*". Biological Cybernetics.

Bhatia, K. (1994).“*Genetic Algorithms and the Traveling Salesman Problem*". Citeseerx. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.9819&rep=rep1&type=pdf viewed 2017

Branimir Sigl, Marin Golub, Vedran Mornar (2003). “*Solving Timetable Scheduling Problem by Using Genetic Algorithms*". Faculty of Electrical Engineering and Computing, University of Zagreb Unska 3, 10000 Zagreb, Croatia.

Burke, E., Kendall, G. (2005).“*Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*". Introduction. In Burke, E., Kendall, G.(eds.), Springer, pp. 5-18.

Can Berk KALAYCI, Aşkıner GÜNGÖR (2011). “*A Genetic Algorithm Based Examination Timetabling Model Focusing on Student Success for the Case of the College of Engineering at Pamukkale University*". Turkey. Gazi University Journal of Science.

Davis L. (1991). “*Handbook of Genetic Algorithms*" . New York, Van Nostrand Reinhold.

E. K. Burke and J. P. Newall (1999). “*A multistage evolutionary algorithm for the timetable problem*". IEEE Transactions on Evolutionary Computation, 3(1): 63–74.

Alaraibi E. (2012). “*Solving timetabling problem by Genetic algorithm*". Dept. of Computer Science, University of Benghazi, Libya.

Fadi A. Aloul, Syed Z. H. Zahidi, Anas Al-Farra, Basel Al-Roh (2013). “*Solving the Employee Timetabling Problem Using Advanced SAT & ILP Techniques*". JOURNAL OF COMPUTERS, VOL. 8, NO. 4.

Fong, C., Asmuni, H., McCollum, B. (2015). “*A Hybrid Swarm-Based Approach to University Timetabling*". Transactions of evolutionary computation. 19(6) pp. 870-884.

Fong, C,. Asmuni, H., McCollum, B., McMullan, P., Omatu, S. (2014). “*A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems*". New Trend of Computational Intelligence in Human-Robot Interaction. 283 pp. 1-21.DOI:10.1016/j.ins.2014.05.039.

Goldberg D.E. and Lingle R. (1985). “*Alleles, loci and the travelling salesman problem*" . Proceedings of an International Conference on Genetic Algorithms, Morgan Kauffman, pp 10-19.

H. M. Sani and M. M. Yabo (2016). “*Solving Timetabling problems using Genetic Algorithm Technique*". International Journal of Computer Applications.

Holland, J. (1975). “*Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology*", Control, and Artificial Intelligence. University of Michigan Press, Ann Arbor.

HyQu, R., Burke, E.K., McCollum, B., Merlot, L.T.G. and Lee, S.Y. (2009). “*A Survey of Search Methodologies and Automated System Development for Examination Timetabling*". Journal of Scheduling. 12(1):: 55-89.

Khairul Anwar, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, Mohammed A.Awadallah (2013). “*Harmony Search-based Hyperheuristic for Examination Timetabling* ".2013 IEEE 9th International Colloquium on Signal Processing and its Applications. Kuala Lumpur, Malaysia.

Marko ýupiü, Marin Golub, Domagoj Jakoboviü (2009). “ *Exam Timetabling Using Genetic Algorithm*". Proceedings of the 31st International Conference on Information Technology Interfaces. Zagreb : University Computing Centre.

MIN WANG (2012). “*Description and Application of Genetic Algorithm*". Blekinge Institute of Technology.

Michalewicz, Z. (1992). “*Genetic Algorthim + Data Structure = Evolution Programs*", Springer Verlag, Berlin Heidelberg.

M. R. Garey and D. S. Johnson (1979). “*Computers and Intractability - A guide to NPcompleteness*", First ed. San Francisco: W. H. Freeman and Company.

Ms.Jayashree Rankhambe , Mrs.S.Kavita (2014). “*Optimization of Examination Timetable Using Harmony Search Hyper-Heuristics (HSHH)*" . International Journal of Computer Science and Information Technologies.

M. W. Carter and G. Laporte (1998). “*Recent developments in practical course timetabling*". Proceedings of the 2nd International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science 1408, pp. 3–19.

Oliver I.M., Smith D.J. and Holland J.H. (1987). “*A study of permutation crossover operators on the travelling salesman problem*". In Proceedings of the Third International Conference on Genetic Algorithms, London, Lawrence Eribaum Associates.

Oluwasefunmi T. Arogundade, Adio T. Akinwale and Omotoyosi M. Aweda (2010). “*A Genetic Algorithm Approach for a Real-World University Examination Timetabling Problem".* Article in International Journal of Computer Applications. (0975 – 8887) Volume 12– No.5.

Petrovic, S., Burke, B. (2004). “*University Timetabling*". In Leung, J. Handbook of Scheduling: Algorithms, Models and Performance Analysis. CRC Press. Chapter 45.

Pillay, P. (2014). “*A survey of school timetabling research*". Annals of operations research. 218(1) pp. 261-293. DOI:10.1007/s10479-013-1321-8.

Pinedo, M.L. (2008). “*Scheduling Theory, Algorithms, and Systems*". Springer.

P. LARRAN AGA, C.M.H. KUIJPERS, R.H. MURGA, I. INZA and S. DIZDAREVIC (1999). “*Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators*" , Dept. of Computer Science and Artificial Intelligence, University of the Basque Country. Spain, Donostia.

Qu, R, and Burke E.K. (2007). “*Adaptive Decomposition and Construction for Examination Timetabling Problems*". Proceedings of the 3rd Multidisciplinary International Scheduling: Theory and Applications, 418-425, Paris, France.

Rakesh Kumar, Girdhar Gopal, Rajesh Kumar (2013). “*Novel Crossover Operator for Genetic Algorithm for Permutation Problems*" . International Journal of Soft Computing and Engineering (IJSCE). ISSN: 2231-2307, Volume-3, Issue-2.

R. Balakrishnan , K. Ranganathan (2012). “*Basic Results: A Textbook of Graph Theory*". Springer New York Heidelberg Dordrecht London.

Reinhard Diestel (2000). “*Graph Theory* ",Springer-Verlag New York.

Rich DC. (1995). *"A Smart Genetic Algorithm for University Timetabling"*. In Burke E and Ross P (eds): Lecture Notes in Computer Science 1153 Practice and Theory of Automated Timetabling First International Conference, Edinburgh, U.K., August/September 1995, Selected Papers. New York: Springer-Verlag Berlin Heidelberg.

R. Lewis (2008). *"A survey of metaheuristic based techniques for university timetabling problems"*. Operation Research Spectrum, 30(1): 167–190.

R. Lewis (2006). *"Metaheuristics for university course timetabling"*. Ph.D. Thesis,School of Computing, Napier University, Edinburgh.

Rushil Raghavjee, Nelishia Pillay (2010). *"An Informed Genetic Algorithm for the High School Timetabling Problem"*. SAICSIT '10 Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists. Bela Bela, South Africa.

Sadaf Naseem Jat (2012). *"Genetic Algorithms for University Course Timetabling Problems"*. Department of Computer Science,University of Leicester.

S. Even, A. Itai, and A. Shamir (1976). *"On the complexity of timetable and multicommodity flow problems"*. SIAM Journal on Computing, 5(4): 691–703.

Syariza Abdul Rahman, Andrzej Bargiela (2014). *"Adaptive linear combination of heuristic orderings in constructing examination timetables"*. European Journal of Operational Research 232, pp. 287–297.

TEOH CHONG KEAT (2012). *"OPTIMIZATION OF ACADEMIC TIMETABLING USING GENETIC ALGORITHM"*. Faculty of Computer Science and Information System,Universiti Teknologi Malaysia JANUARY.

Twinkle Dahiya, Er. Kapil, Er.Aakash Goyal (2015). *"Exam Timetabling Problem Using G.A"*. Twinkle Dahiya al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 2, Issue 2, pp. 165-168.

Walid Y. Abdel Wahed (2011). *"Solving time table problem using genetic algorithm"*. Department of Computer Science, Academy of graduate Studies. Benghazi – Libya.

W. Thomas (2009). *"Global Optimization Algorithms Theory and Application"*. Thomas Weise.

Younis R. Elhaddad and Aiman S.Gannous (2010).*" New crossover technique for genetic algorithm"*. 2010 International Conference on Computer and Software Modeling (ICCSM 2010) . 978-1-4244-9097 -4/10/$26.00©2010 IEEE.

Y. Elhaddad, O. Sallabi (2009). *"Anew Hybrid Genetic and simulated annealing algorthim to solve the trvaling selesman problem"*, Department of Computer Science,University of Benghazi.

Appendices

APPENDIX A

FACULTY OF INFORMATION TECHNOLOGY UNDERGRADUATES COURSES

Program: Bachelor of Computer Science

Year 1

Semester I

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT101	IT Foundations	3	--	--
IT111	Programming I	4	--	--
GE111	English I	3	--	--
GS113	Math I	3	--	--
GS221	Probability and Statistics	3	--	--

Semester II

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT112	Programming II	4	IT111	Programming I
IT121	Digital and Logic Design	3	IT101	IT Foundations
GE112	English II	3	GE111	English I
GS114	Math II	3	GS113	Math I
GE101	Arabic I	3	--	--
GS141	General Physics I + Lab I	4	--	--

Year 2

Semester III

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT212	Data Structures and Algorithms	3	IT112	Programming II
IT201	Discrete Math and Structures	3	IT111 GS113	Programming I Math I
SE201	Foundation of Software Eng.	3	IT112	Programming II
IS201	Foundations of Information Systems	3	IT112	Programming II
GE331	Political Culture	3	--	--

Semester IV

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT222	Computer Architecture	3	IT121	Digital and Logic Design
IT271	Network Fundamentals	3	IT121 IT112	Digital and Logic Design Programming II
CS205	Numerical Analysis	3	GS114 IT112	Math II Programming II
CS211	Analysis of algorithms	3	IT201 T112	Discrete Math and Structures Programming II
GE311	Technical writing	3	GE112	English II

Year 3

Semester V

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT341	Database systems	3	IT212	Data Structures and Algorithms
IT322	Operating systems	3	IT212	Data Structures and Algorithms
CS331	Programming languages	3	IT212	Data Structures and Algorithms
SE322	Human computer interaction	3	SE201	Foundation of Software Eng.
IS361	Systems analysis and design	3	IS201	Foundations of Information Systems

Semester VI

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT342	Security principles and practices	3	IT271	Network Fundamentals
IT301	Computing ethics& society	3	IT101 IT112	IT Foundations Programming II
CS341	Software development	3	IS361 IT341	Systems analysis and design Database systems
CN281	Internet Programming	3	IT271 T112	Network Fundamentals Programming II
	Dept. Elective I	3		

Year 4

Semester VII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT492	Internship	3	90 Cr.Hrs	--
CS451	Intelligent Systems	3	IT212 IT201	Data Structures and Algorithms Discrete Math and Structures.
CS461	Computer Graphics	3	IT112 IT212	Programming II Data Structures and Algorithms
SE492	Software project management	3	SE201	Foundation of Software Eng.
	Dept. Elective II	3		
	Breadth Elective I	3		

Semester VIII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT499	IT Capstone Project	4	100 Cr. rs	--
CS442	Advanced database	3	IT341	Database systems
CS462	Software Multimedia Technology	3	IT322	Operating systems
	Dept. Elective III	3		
	Breadth Elective II	3		

Program: Bachelor of Software Engineering

Year 1

Semester I

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT101	IT Foundations	3	--	--
IT111	Programming I	4	--	--
GE111	English I	3	--	--
GS113	Math I	3	--	--
GS221	Probability and Statistics	3	--	--

Semester II

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT112	Programming II	4	IT111	Programming I
IT121	Digital and Logic Design	3	IT101	IT Foundations
GE112	English II	3	GE111	English I
GS114	Math II	3	GS113	Math I
GE101	Arabic I	3	--	--
GS141	General Physics I + Lab I	4	--	--

Year 2

Semester III

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT212	Data Structures and Algorithms	3	IT112	Programming II
IT201	Discrete Math and Structures	3	IT111 GS113	Programming I Math I
SE201	Foundation of Software Eng.	3	IT112	Programming II
IS201	Foundations of Information Systems	3	IT112	Programming II
GE331	Political Culture	3	--	--

Semester IV

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT222	Computer Architecture	3	IT121	Digital and Logic Design
IT271	Network Fundamentals	3	IT121 IT112	Digital and Logic Design Programming II
SE211	Software Requirements	3	SE201	Foundation of Software Eng
CS211	Analysis of algorithms	3	IT201 T112	Discrete Math and Structures Programming II
GE311	Technical writing	3	GE112	English II

Year 3

Semester V

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT341	Database systems	3	IT212	Data Structures and Algorithms
IT322	Operating systems	3	IT212	Data Structures and Algorithms
SE312	Formal Models & Methods	3	SE201 IT201	Foundation of Software Eng. Discrete Math and Structures.
SE322	Human computer interaction	3	SE201	Foundation of Software Eng.
SE321	Software Design	3	SE201	Foundation of Software Eng.

Semester VI

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT342	Security principles and practices	3	IT271	Network Fundamentals
IT301	Computing ethics& society	3	IT101 IT112	IT Foundations Programming II
SE331	Software Testing	3	SE201	Foundation of Software Eng.
SE341	Software Evolution and Maintenance	3	SE201	Foundation of Software Eng.
	Dept. Elective I	3		

Year 4

Semester VII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT492	Internship	3	90 Cr.Hrs	--
SE441	Re-use and Component Based Development	3	SE321	Software Design
SE461	Software Quality	3	SE321	Software Design
CN281	Internet Programming	3	IT112	Programming II
	Dept. Elective II	3		
	Breadth Elective I	3		

Semester VIII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT499	IT Capstone Project	4	100 Cr. rs	--
SE490	Software Development	3	IT341 SE321	Database systems Software Design
SE492	Software Project Management	3	SE201	Foundation of Software Eng
	Dept. Elective III	3		
	Breadth Elective II	3		

Program: Bachelor of Information Systems

Year 1

Semester I

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT101	IT Foundations	3	--	--
IT111	Programming I	4	--	--
GE111	English I	3	--	--
GS113	Math I	3	--	--
GS221	Probability and Statistics	3	--	--

Semester II

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT112	Programming II	4	IT111	Programming I
IT121	Digital and Logic Design	3	IT101	IT Foundations
GE112	English II	3	GE111	English I
GS114	Math II	3	GS113	Math I
GE101	Arabic I	3	--	--
GS141	General Physics I + Lab I	4	--	--

Year 2

Semester III

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT212	Data Structures and Algorithms	3	IT112	Programming II
IT201	Discrete Math and Structures	3	IT111 GS113	Programming I Math I
SE201	Foundation of Software Eng.	3	IT112	Programming II
IS201	Foundations of Information Systems	3	IT112	Programming II
GE331	Political Culture	3	--	--

Semester IV

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT222	Computer Architecture	3	IT121	Digital and Logic Design
IT271	Network Fundamentals	3	IT121 IT112	Digital and Logic Design Programming II
IS271	Financial accounting	3	GS113	Math I
CN281	Internet programming	3	IT112	Programming II
GE311	Technical writing	3	GE112	English II

Year 3

Semester V

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT341	Database systems	3	IT212	Data Structures and Algorithms
IT322	Operating systems	3	IT212	Data Structures and Algorithms
IS321	IS theory and practices	3	IS201 IT201	Foundations of IS. Discrete Math and Structures.
IS351	Organizational behaviour	3	IS201	Foundations of IS.
IS361	Systems analysis and design	3	IS201	Foundations of IS.

Semester VI

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT342	Security principles and practices	3	IT271	Network Fundamentals
IT301	Computing ethics& society	3	IT101 IT112	IT Foundations Programming II
SE322	Human computer interaction	3	SE201	Foundation of Software Eng.
IS362	Information systems development	3	IS361	Systems analysis and design
	Dept. Elective I	3		

Year 4

Semester VII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT492	Internship	3	90 Cr.Hrs	--
IS372	Enterprise resource planning	3	IS201	Foundations of IS.
SE321	Software design	3	SE201	Foundations of software eng.
IS475	E-Commerce and E-Marketing	3	IS201 CN281	Foundations of IS. Internet programming
	Dept. Elective II	3		
	Breadth Elective I	3		

Semester VIII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT499	IT Capstone Project	4	100 Cr.rs	--
CS451	Intelligent systems	3	IT201 IT212	Discrete Math and Structures Data structures and algorithms
SE492	Software Project Management	3	SE201	Foundation of Software Eng.
	Dept. Elective III	3		
	Breadth Elective II	3		

Program: Bachelor of Computer Networks and Communications

Year 1

Semester I

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT101	IT Foundations	3	--	--
IT111	Programming I	4	--	--
GE111	English I	3	--	--
GS113	Math I	3	--	--
GS221	Probability and Statistics	3	--	--

Semester II

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT112	Programming II	4	IT111	Programming I
IT121	Digital and Logic Design	3	IT101	IT Foundations
GE112	English II	3	GE111	English I
GS114	Math II	3	GS113	Math I
GE101	Arabic I	3	--	--
GS141	General Physics I + Lab I	4	--	--

Year 2

Semester III

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT212	Data Structures and Algorithms	3	IT112	Programming II
IT201	Discrete Math and Structures	3	IT111 GS113	Programming I Math I
CN261	Electric basics	3	GS141	General Physics I+Lab I
CN281	Internet Programming	3	IT112	Programming II
GE331	Political Culture	3	--	--

Semester IV

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT222	Computer Architecture	3	IT121	Digital and Logic Design
IT271	Network Fundamentals	3	IT121 IT112	Digital and Logic Design Programming II
IS271	Financial accounting	3	GS113	Math I
CN262	Theory of Signals and Systems	3	CN261	Electric basics
CN271	Intro. to Stochastic & Random process	3	GS221	Probability and Statistics

Year 3

Semester V

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT341	Database systems	3	IT212	Data Structures and Algorithms
IT322	Operating systems	3	IT212	Data Structures and Algorithms
CN321	Network Protocols	3	IT271	Network Fundamentals
CN311	Networks Lab 1	1	IT271	Network Fundamentals
CN382	Advanced Internet Programming	3	CN281	Internet Programming
CN371	Digital Communications	3	CN262 CN271	Theory of Signals and Systems Intro. to Stochastic & Random process

Semester VI

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT342	Security principles and practices	3	IT271	Network Fundamentals
IT301	Computing ethics& society	3	IT101 IT112	IT Foundations Programming II
CN331	Wireless and Mobile Communications	3	CN371 CN321	Digital Communications Network Protocols
CN322	Network lab 2	1	CN311 CN321	Networks Lab 1 Network Protocols
CN312	Network operating systems software	3	IT322 CN311	Operating systems Networks Lab 1
	Dept. Elective I	3		

Year 4

Semester VII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT492	Internship	3	90 Cr.Hrs	--
CN498	Introduction to Project	3	90 Cr.Hrs	--
CN483	Mobile Commerce	3	IT342 CN382	Security principles and practices Advanced Internet Programming
CN473	Networks Lab 3	1	CN322	Network lab 2
	Dept. Elective II	3		
	Breadth Elective I	3		

Semester VIII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT499	IT Capstone Project	4	100 Cr.rs	--
CN413	Network Planning and Deployment	3	CN321 CN473	Network Protocols Networks Lab 3
CN441	Network Security	3	CN322 IT342	Network lab 2 Security principles and practices
CN484	Mobile Computing	3	CN382 CN483	Advanced Internet Programming Mobile Commerce
	Dept. Elective III	3		
	Breadth Elective II	3		

Program: Bachelor of Computer Systems Design

Year 1

Semester I

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT101	IT Foundations	3	--	--
IT111	Programming I	4	--	--
GE111	English I	3	--	--
GS113	Math I	3	--	--
GS221	Probability and Statistics	3	--	--

Semester II

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT112	Programming II	4	IT111	Programming I
IT121	Digital and Logic Design	3	IT101	IT Foundations
GE112	English II	3	GE111	English I
GS114	Math II	3	GS113	Math I
GE101	Arabic I	3	--	--
GS141	General Physics I + Lab I	4	--	--

Year 2

Semester III

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT212	Data Structures and Algorithms	3	IT112	Programming II
IT201	Discrete Math and Structures	3	IT111 GS113	Programming I Math I
IT271	Network Fundamentals	3	IT121	General Physics I+Lab I
IT222	Computer Architecture	3	IT121	Digital and Logic Design
CE201	Circuits Fundamentals	3	GS141	
GE331	Political Culture	3	--	--

Semester IV

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
CE200	Computer Architecture& implementation.	3	IT222	Computer Architecture
CE202	Digital electronics	3	CE201	Circuits Fundamentals
CE203	Digital electronics lab	1	CE201	Circuits Fundamentals
CE210	Advanced digital design with HDL	3	IT222	Computer Architecture
CE211	Advanced digital design with HDL lab	1	IT222	Computer Architecture
CE221	Signal and System	3	IT222	Computer Architecture
GE311	Technical writing	3	GE112	English II

Year 3

Semester V

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT341	Database systems	3	IT212	Data Structures and Algorithms
IT322	Operating systems	3	IT212	Data Structures and Algorithms
CE220	Embedded System	3	IT222 CE201	Computer Architecture Circuits Fundamentals
CE311	Microcomputer	3	CE201 CE202	Circuits Fundamentals Digital electronics
CE310	Microcomputer lab	1	CE201 CE203	Circuits Fundamentals Digital electronics lab
CE321	Digital Signal Processing	3	--	--
CE300	Platform Architecture and technologies	3	CE201	Circuits Fundamentals

Semester VI

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT342	Security principles and practices	3	IT271	Network Fundamentals
IT301	Computing ethics& society	3	IT101 IT112	IT Foundations Programming II
CE312	Microcontroller System Design	3	CE311	Microcomputer
CE313	Microcontroller System Design lab	1	CE311	Microcomputer
CE325	Hardware Testing & Fault Tolerance	3	IT222	Computer Architecture
	Dept. Elective I	3		

Year 4

Semester VII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT492	Internship	3	90 Cr.Hrs	--
CS451	Intelligent system	3	IT201 IT212	Discrete Math and Structures Data structures and algorithms
	Dept. Elective II	3		
	Dept. Elective III	3		
	Breadth Elective I	3		

Semester VIII

CODE	COURSE	CREDITS	PRE-REQUISITE	
			CODE	COURSE
IT499	IT Capstone Project	4	100 Cr.rs	--
CE499	Special Topics in computer system	3	--	--
	Dept. Elective V	3		
	Breadth Elective II	3		

APPENDIX B

LIST OF LECTURERS

Teacher			
TeachersNo	EmployeeID	TeacherName	StatusTeachers
1	111	د.اشرف	قار
2	112	أ.جعفر	متعاون
3	113	أ.أمينة	قار
4	114	أ. المجبري	قار
5	115	أ.حاتم	قار
6	116	أ.تغريد	متعاون
7	117	أ.بلال	قار
8	118	أ.المنصوري	متعاون
9	119	د.الطويل	قار
10	120	أ.رانيا ح	متعاون
11	121	أ.الفيتوري	قار
12	122	أ.الهوني	قار
13	123	أ. القلالي	قار
14	124	أ.الساحلي	متعاون
15	125	أ.خديجة	قار
16	126	أ.الفرجاني	قار
17	127	أ.ياسين	متعاون
18	128	أ.جودة	قار
19	129	د.رجب	قار
20	130	أ.جدولة	متعاون
21	131	أ.زكية	قار
22	132	أ.سالمة	قار
23	133	أ.بوعزة	قار
24	134	ا.سلوى	قار
25	135	أ.سالم	قار
26	136	أ.بودبس	قار
27	137	د.شعبان	قار
28	138	د. شارما	قار
29	139	أ.صالح	قار
30	140	د.أشهيبية	قار
31	141	أ.طارق الناجى	قار
32	142	أ.عزالدين	متعاون
33	143	د.عبد الواحد	قار
34	144	د.علي	قار
35	145	د.أعنيبة	قار
36	146	د.عبد الكافى	قار
37	147	أ.خليل	قار
38	148	د.عمر	قار
39	149	أ.الدرسي	قار
40	150	د.المؤدب	قار

Teacher			
TeachersNo	EmployeeID	TeacherName	StatusTeachers
41	151	د.الطبولي	قار
42	152	أ.فراس	قار
43	153	أ.فتحية	قار
44	154	د.كنز	قار
45	155	أ.مها	قار
46	156	د.بالحسن	قار
47	157	أ.الزوى	قار
48	158	أ.مروان	قار
49	159	د.منى	قار
50	160	أ.نوح	قار
51	161	د.م سالم	قار
52	162	أ.حجل	قار
53	163	د.اخليف	قار
54	164	أ.العرفى	قار
55	165	أ.الشاعرى	قار
56	166	أ.الزغيد	قار
57	167	أ.بورويص	قار
58	168	أ.البرغثى	قار
59	169	أ.السعيطى	قار
60	170	أ.ونيسة	قار
61	171	د.المقصبي	قار
62	172	د.ياسمينه	قار
63	173	أ.الحداد	قار
64	174	د.الشكماك	قار
65	175	أ.امعيتيق	قار
66	176	غ محدد	قار
67	177	غ محدد	قار
68	178	غ محدد	قار
69	179	غ محدد	قار
70	180	غ محدد	قار
71	181	غ محدد	قار
72	182	غ محدد	قار
73	183	أ. ككلي	متعاون
74	184	أ. أحميدة	متعاون
75	185	أ.ايناس	متعاون
76	186	أ. غادة	قار
77	187	أ.رافع	متعاون
78	188	أ.سالم ف	متعاون
79	189	أ. الكيلانى	متعاون
80	190	أ. الدرولى	متعاون

Teacher			
TeachersNo	EmployeeID	TeacherName	StatusTeachers
81	191	أ.فوزى	متعاون
82	192	أ.إسلام	متعاون
83	193	أ.بن صريتي	متعاون
84	194	أ. سامي	متعاون
85	195	أ.وسام بن	متعاون
86	196	أ. المصري	متعاون
87	197	أ. هلال	متعاون
88	198	أ. البشارى	متعاون
89	199	أ.ضوء	قار
90	200	أ. الرملي	قار
91	201	أ. نبيلة	قار
92	202	أ. رشا	قار
93	203	أ.بالحاج	قار
94	204	أ.الشلماني	متعاون
95	205	أ. الجهانى	متعاون
96	206	أ. المطردي	متعاون
97	207	أ. هدية	متعاون
98	208	أ. رانيا	متعاون
99	209	أ. الغزالي	متعاون
100	210	د. أحمد اوج	متعاون
101	211	أ. حازم	متعاون
102	212	أ. وليد	متعاون
103	213	أ. القطعانى	متعاون
104	214	أ. ماضي	متعاون
105	215	أ.بسمة	متعاون
106	216	أ.سوسية	متعاون
107	217	أ.رمضان ط	متعاون
108	218	أ. هند	متعاون
109	219	أ. باكير	متعاون
110	220	أ.الهام با	متعاون
111	221	أ. الاثرم	متعاون
112	222	أ. معمر	متعاون
113	223	أ. الراشدي	متعاون
114	224	أ.الترهوني	متعاون
115	225	أ. الحوتي	متعاون
116	226	أ. مخزوم	قار
117	227	أ. حنان م	قار
118	228	أ. تامر	قار
119	229	أ.راشون	متعاون
120	230	أ. سعيد	متعاون

Teacher			
TeachersNo	EmployeeID	TeacherName	StatusTeachers
121	231	أ. الجملى	متعاون
122	232	أ. بادي	قار
123	233	أ. هنيدة	قار
124	234	أ. الفقيه	قار
125	235	أ. الصيد	قار
126	236	أ. نجاة	قار
127	237	أ. الشويب	قار
128	238	أ. قرش	متعاون
129	239	أ. تهاني د	متعاون
130	240	أ.سارية	قار
131	241	أ.زهدي	متعاون
132	242	أ.العريبي	فار
133	243	أ. البركاوى	فار
134	244	أ. البابور	قار
135	245	أ. معتوق	قار
136	246	أ. الاوجلي	قار
137	247	د. الشلماني	قار
138	248	د. م مختار	متعاون
139	249	أ. الكوافي	متعاون
140	250	أ. العباني	متعاون
141	251	أ. خ المقصبي	متعاون
142	252	أ. جعفر	متعاون
143	253	د. عبدالكافي	متعاون
144	254	أ.الصادق	متعاون
145	255	أ.مكراز	متعاون
146	256	منتدب IT	متعاون
147	257	IT112 1م	متعاون
148	258	أ. بسمة م	متعاون
149	260	أ. فركاش	متعاون
150	259	أ.الهام حاس	متعاون
151	261	أ. الشريف	متعاون
152	262	IT201 1 م	متعاون

APPENDIX C

LIST OF ROOMS

Room			
SNR	RoomName	Capacity	Type
1	قاعه 9	40	Classroom
2	قاعة 10	35	Classroom
3	قاعة 103	35	Classroom
4	قاعة 104	30	Classroom
5	قاعة 205	30	Classroom
6	قاعة 208	30	Classroom
7	قاعة 244	30	Classroom
8	قاعة 106	50	Classroom
9	قاعة 105	50	Classroom
10	قاعة 316	40	Classroom
11	مدرج ق 1	100	Classroom
12	مدرج ق 2	100	Classroom
13	Lab A	30	Lab
14	Lab B	30	Lab
15	Lab C	40	Lab
16	Lab G	40	Lab

APPENDIX D

LIST OF LECTURES

Query1							
SN	Cours_Number	ECoursName	GroupName	Credits	TeacherName	RoomName	Capacity
1	SE201	Foundation of Software Engineering	A	3	أ.تغريد	قاعة 103	35
2	SE201	Foundation of Software Engineering	A	3	أ.تغريد	قاعة 103	35
3	SE201	Foundation of Software Engineering	B	3	أ.تغريد	قاعة 103	35
4	SE201	Foundation of Software Engineering	B	3	أ.تغريد	قاعة 103	35
5	SE201	Foundation of Software Engineering	C	3	أ.تغريد	قاعة 103	35
6	SE201	Foundation of Software Engineering	C	3	أ.تغريد	قاعة 103	35
7	SE211	Software Requirements	A	3	أ.بلال	قاعة 104	30
8	SE211	Software Requirements	A	3	أ.بلال	قاعة 104	30
9	SE211	Software Requirements	B	3	أ.المنصوري	قاعة 104	30
10	SE211	Software Requirements	B	3	أ.المنصوري	قاعة 104	30
11	SE301	Engineering Economics	A	3	أ.رانيا ح	قاعة 103	35
12	SE301	Engineering Economics	A	3	أ.رانيا ح	قاعة 103	35
13	SE312	Formal Methods and Methods	A	3	أ. القلالي	قاعة 104	30
14	SE312	Formal Methods and Methods	A	3	أ. القلالي	قاعة 104	30
15	SE321	Software Design	A	3	أ.الساحلي	قاعة 103	35
16	SE321	Software Design	A	3	أ.الساحلي	قاعة 103	35
17	SE321	Software Design	B	3	أ.المنصوري	قاعة 103	35
18	SE321	Software Design	B	3	أ.المنصوري	قاعة 103	35
19	SE321	Software Design	C	3	أ.المنصوري	قاعة 103	35
20	SE321	Software Design	C	3	أ.المنصوري	قاعة 103	35
21	SE322	Human Computer Interaction	A	3	أ.ياسين	قاعة 104	30
22	SE322	Human Computer Interaction	A	3	أ.ياسين	قاعة 104	30
23	SE322	Human Computer Interaction	B	3	أ.ياسين	قاعة 104	30
24	SE322	Human Computer Interaction	B	3	أ.ياسين	قاعة 104	30
25	SE322	Human Computer Interaction	C	3	أ.ياسين	قاعة 104	30
26	SE322	Human Computer Interaction	C	3	أ.ياسين	قاعة 104	30
27	SE331	Software Testing	A	3	أ.بلال	قاعة 103	35
28	SE331	Software Testing	A	3	أ.بلال	قاعة 103	35
29	SE341	Software Evolution and Maintenance	A	3	أ.رانيا ح	قاعة 103	35
30	SE341	Software Evolution and Maintenance	A	3	أ.رانيا ح	قاعة 103	35
31	SE441	Reuse and Component-based Development	A	3	أ.بلال	قاعة 104	30

Query1							
SN	Cours_Number	ECoursName	GroupName	Credits	TeacherName	RoomName	Capacity
32	SE441	Reuse and Component-based Development	A	3	أ.بلال	قاعة 104	30
33	SE461	Software Quality	A	3	أ.رانيا ح	قاعة 104	30
34	SE461	Software Quality	A	3	أ.رانيا ح	قاعة 104	30
35	SE490	Software Development	A	3	أ.رانيا ح	قاعة 103	35
36	SE490	Software Development	A	3	أ.رانيا ح	قاعة 103	35
37	SE491	Special Topics in Software Engineering	A	3	أ.جدولة	قاعة 103	35
38	SE491	Special Topics in Software Engineering	A	3	أ.جدولة	قاعة 103	35
39	SE492	Software project managment	A	3	أ.بوعزة	قاعة 104	30
40	SE492	Software project managment	A	3	أ.بوعزة	قاعة 104	30
41	SE492	Software project managment	B	3	أ.بوعزة	قاعة 104	30
42	SE492	Software project managment	B	3	أ.بوعزة	قاعة 104	30
43	SE492	Software project managment	C	3	أ.بوعزة	قاعة 104	30
44	SE492	Software project managment	C	3	أ.بوعزة	قاعة 104	30
45	CS205	Numerical Analysis	A	3	أ.سالمة	قاعة 244	30
46	CS205	Numerical Analysis	A	3	أ.سالمة	قاعة 244	30
47	CS205	Numerical Analysis	B	3	أ.سالمة	قاعة 244	30
48	CS205	Numerical Analysis	B	3	أ.سالمة	قاعة 244	30
49	CS211	Analysis of algoirthms	B	3	أ.الفيتوري	قاعة 244	30
50	CS211	Analysis of algoirthms	B	3	أ.الفيتوري	قاعة 244	30
51	CS211	Analysis of algoirthms	A	3	أ.الفيتوري	قاعة 244	30
52	CS211	Analysis of algoirthms	A	3	أ.الفيتوري	قاعة 244	30
53	CS331	Programming languages	A	3	أ. الغزالي	قاعة 244	30
54	CS331	Programming languages	A	3	أ. الغزالي	قاعة 244	30
55	CS331	Programming languages	B	3	أ. الغزالي	قاعة 244	30
56	CS331	Programming languages	B	3	أ. الغزالي	قاعة 244	30
57	CS341	Software development	A	3	أ.المنصوري	قاعة 244	30
58	CS341	Software development	A	3	أ.المنصوري	قاعة 244	30
59	CS431	Compiler Construction	A	3	أ.سارية	قاعة 244	30
60	CS431	Compiler Construction	A	3	أ.سارية	قاعة 244	30
61	CS442	Advanced Database	A	3	د.المؤدب	قاعة 244	30
62	CS442	Advanced Database	A	3	د.المؤدب	قاعة 244	30
63	CS443	Data and Web Mining	A	3	د.المؤدب	قاعة 244	30
64	CS443	Data and Web Mining	A	3	د.المؤدب	قاعة 244	30
65	CS451	Intelligent Systems	A	3	د.أشهيبية	قاعة 244	30
66	CS451	Intelligent Systems	A	3	د.أشهيبية	قاعة 244	30
67	CS451	Intelligent Systems	B	3	د.أشهيبية	قاعة 244	30
68	CS451	Intelligent Systems	B	3	د.أشهيبية	قاعة 244	30

Query1							
SN	Cours_Number	ECoursName	GroupName	Credits	TeacherName	RoomName	Capacity
69	CS451	Intelligent Systems	C	3	أ.خديجة	قاعة 244	30
70	CS451	Intelligent Systems	C	3	أ.خديجة	قاعة 244	30
71	CS461	Computer Graphics	A	3	أ. القلالي	قاعة 244	30
72	CS461	Computer Graphics	A	3	أ. القلالي	قاعة 244	30
73	CS462	Multimedia Technology	A	3	د.أشهيبية	قاعة 244	30
74	CS462	Multimedia Technology	A	3	د.أشهيبية	قاعة 244	30
75	CS462	Multimedia Technology	B	3	أ.زهدي	قاعة 244	30
76	CS462	Multimedia Technology	B	3	أ.زهدي	قاعة 244	30
77	CS471	Distributed Systems	A	3	أ.زهدي	قاعة 244	30
78	CS471	Distributed Systems	A	3	أ.زهدي	قاعة 244	30
79	IS201	Foundation of Information Systems	A	3	أ.وسام بن	قاعة 205	30
80	IS201	Foundation of Information Systems	A	3	أ.وسام بن	قاعة 205	30
81	IS201	Foundation of Information Systems	B	3	أ.وسام بن	قاعة 205	30
82	IS201	Foundation of Information Systems	B	3	أ.وسام بن	قاعة 205	30
83	IS201	Foundation of Information Systems	C	3	د.عمر	قاعة 205	30
84	IS201	Foundation of Information Systems	C	3	د.عمر	قاعة 205	30
85	IS271	Financial Accounting	A	3	د.عبد الكافي	قاعة 208	30
86	IS271	Financial Accounting	A	3	د.عبد الكافي	قاعة 208	30
87	IS271	Financial Accounting	B	3	د.عبد الكافي	قاعة 208	30
88	IS271	Financial Accounting	B	3	د.عبد الكافي	قاعة 208	30
89	IS321	IS Theory and Practices	A	3	أ. البابور	قاعة 205	30
90	IS321	IS Theory and Practices	A	3	أ. البابور	قاعة 205	30
91	IS321	IS Theory and Practices	B	3	أ. البابور	قاعة 205	30
92	IS321	IS Theory and Practices	B	3	أ. البابور	قاعة 205	30
93	IS351	Organizational behavior	A	3	أ. البابور	قاعة 208	30
94	IS351	Organizational behavior	A	3	أ. البابور	قاعة 208	30
95	IS361	Systems analysis & design	A	3	أ. معتوق	قاعة 205	30
96	IS361	Systems analysis & design	A	3	أ. معتوق	قاعة 205	30
97	IS361	Systems analysis & design	B	3	أ. معتوق	قاعة 205	30
98	IS361	Systems analysis & design	B	3	أ. معتوق	قاعة 205	30
99	IS362	Information Systems Development	A	3	أ.وسام بن	قاعة 208	30
100	IS362	Information Systems Development	A	3	أ.وسام بن	قاعة 208	30
101	IS374	Principles of Marketing	A	3	د. أحمد اوج	قاعة 208	30

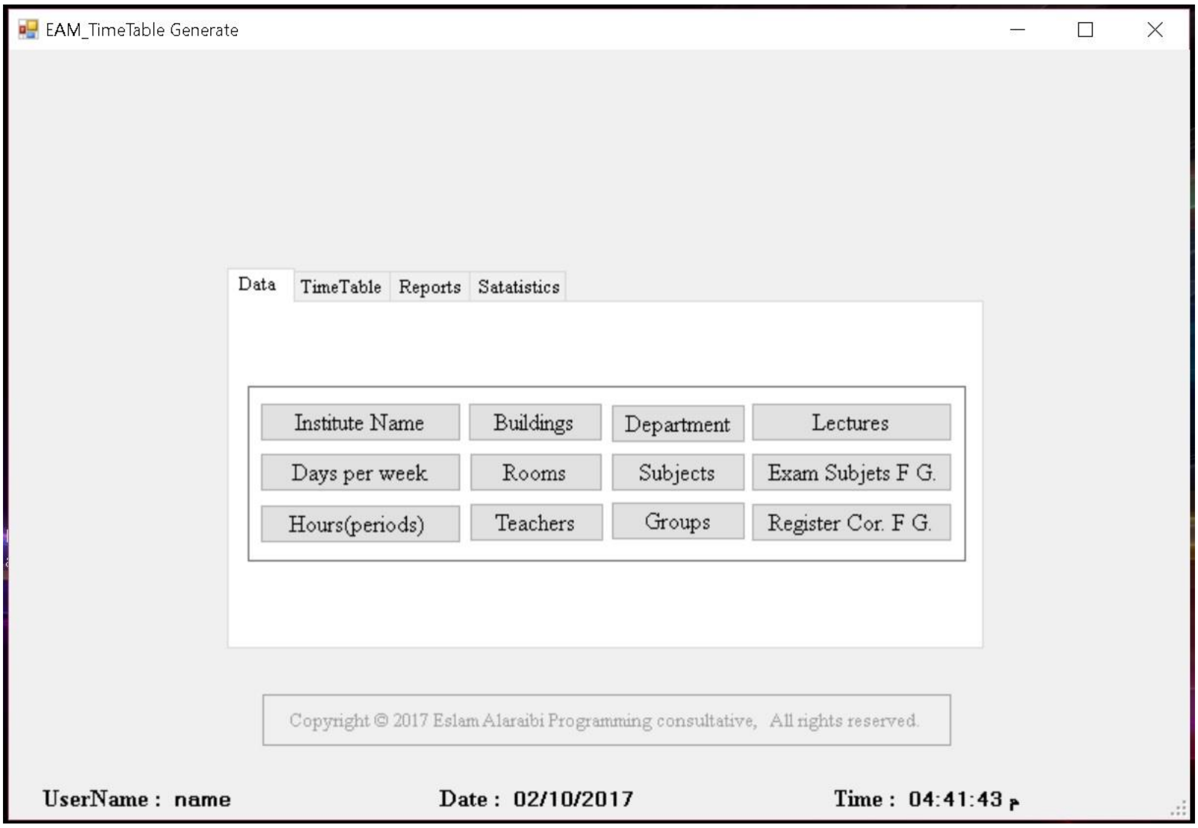
Query1							
SN	Cours_Number	ECoursName	GroupName	Credits	TeacherName	RoomName	Capacity
.	IT111	Programming I	F	4	أ. الغزالي	قاعة 106	50
.	IT111	Programming I	F	4	أ. الغزالي	قاعة 106	50
.	IT111	Programming I	G	4	أ. حازم	قاعة 105	50
.	IT111	Programming I	G	4	أ. حازم	قاعة 105	50
.	IT111	Programming I	H	4	أ. وليد	قاعه 9	40
.	IT111	Programming I	H	4	أ. وليد	قاعه 9	40
.	IT111	Programming I	I	4	أ. الاوجلي	قاعة 10	35
.	IT111	Programming I	I	4	أ. الاوجلي	قاعة 10	35
.	IT111	Programming I	J	4	أ. ككلي	قاعه 9	40
.	IT111	Programming I	J	4	أ. ككلي	قاعه 9	40
.	IT111	Programming I	K	4	أ.خديجة	قاعة 106	50
.	IT111	Programming I	K	4	أ.خديجة	قاعة 105	50
.	IT112	Programming II (OOP)	A	4	أ. القطعانى	قاعة 10	35
.	IT112	Programming II (OOP)	A	4	أ. القطعانى	قاعة 10	35
.	IT112	Programming II (OOP)	B	4	أ. القطعانى	قاعه 9	40
.	IT112	Programming II (OOP)	B	4	أ. القطعانى	قاعه 9	40
.	IT112	Programming II (OOP)	C	4	أ. ماضي	قاعة 10	35
.	IT112	Programming II (OOP)	C	4	أ. ماضي	قاعة 10	35
.	IT112	Programming II (OOP)	D	4	أ. ماضي	قاعه 9	40
.	IT112	Programming II (OOP)	D	4	أ. ماضي	قاعه 9	40
.	IT112	Programming II (OOP)	E	4	أ.سوسية	قاعة 10	35
.	IT112	Programming II (OOP)	E	4	أ.سوسية	قاعه 9	40
.	IT112	Programming II (OOP)	F	4	أ. أحميدة	قاعة 10	35
.	IT112	Programming II (OOP)	F	4	أ. أحميدة	قاعة 10	35
.	IT112	Programming II (OOP)	G	4	أ. أحميدة	قاعه 9	40
.	IT112	Programming II (OOP)	G	4	أ. أحميدة	قاعة 10	35
.	IT112	Programming II (OOP)	H	4	أ. القلالي	قاعة 105	50
.	IT112	Programming II (OOP)	H	4	أ. القلالي	قاعة 106	50
.	IT112	Programming II (OOP)	I	4	أ.الساحلي	قاعة 105	50
.	IT112	Programming II (OOP)	I	4	أ.الساحلي	قاعة 105	50
.	IT112	Programming II (OOP)	J	4	م1 IT112	قاعة 105	50
.	IT112	Programming II (OOP)	J	4	م1 IT112	قاعة 106	50
.	IT121	Digital Logic Design	A	3	أ. بسمة م	قاعة 316	40
.	IT121	Digital Logic Design	A	3	أ. بسمة م	قاعة 316	40
.	IT121	Digital Logic Design	B	3	أ. بسمة م	قاعة 316	40
.	IT121	Digital Logic Design	B	3	أ. بسمة م	قاعة 316	40
.	IT121	Digital Logic Design	C	3	أ.الهام با	قاعة 316	40
.	IT121	Digital Logic Design	C	3	أ.الهام با	قاعة 316	40
.	IT121	Digital Logic Design	D	3	أ.الهام با	قاعة 316	40
.	IT121	Digital Logic Design	D	3	أ.الهام با	قاعة 316	40

Query1							
SN	Cours_Number	ECoursName	GroupName	Credits	TeacherName	RoomName	Capacity
213	IT121	Digital Logic Design	E	3	أ. الاثرم	قاعة 316	40
214	IT121	Digital Logic Design	E	3	أ. الاثرم	قاعة 316	40
215	IT121	Digital Logic Design	F	3	أ. الاثرم	قاعة 316	40
216	IT121	Digital Logic Design	F	3	أ. الاثرم	قاعة 316	40
217	IT121	Digital Logic Design	G	3	أ. الشريف	قاعه 9	40
218	IT121	Digital Logic Design	G	3	أ. الشريف	قاعة 10	35
219	IT121	Digital Logic Design	H	3	أ. الشريف	قاعة 10	35
220	IT121	Digital Logic Design	H	3	أ. الشريف	قاعه 9	40
221	IT121	Digital Logic Design	I	3	أ. فركاش	قاعة 10	35
222	IT121	Digital Logic Design	I	3	أ. فركاش	قاعة 10	35
223	IT121	Digital Logic Design	J	3	أ.الهام حاس	قاعه 9	40
224	IT121	Digital Logic Design	J	3	أ.الهام حاس	قاعه 9	40
225	IT201	Discrete Math and Structures	E	3	أ.خديجة	قاعة 106	50
226	IT201	Discrete Math and Structures	E	3	أ.خديجة	قاعة 105	50
227	IT212	Data structures and algorithms	A	3	أ. غادة	قاعه 9	40
228	IT212	Data structures and algorithms	A	3	أ. غادة	قاعه 9	40
229	IT212	Data structures and algorithms	B	3	أ. غادة	قاعة 316	40
230	IT212	Data structures and algorithms	B	3	أ. غادة	قاعة 316	40
231	IT212	Data structures and algorithms	C	3	أ. غادة	قاعة 316	40
232	IT212	Data structures and algorithms	C	3	أ. غادة	قاعة 316	40
233	IT212	Data structures and algorithms	D	3	أ.رافع	قاعة 316	40
234	IT212	Data structures and algorithms	D	3	أ.رافع	قاعة 316	40
235	IT212	Data structures and algorithms	E	3	أ.رافع	قاعة 10	35
236	IT212	Data structures and algorithms	E	3	أ.رافع	قاعة 10	35
237	IT212	Data structures and algorithms	F	3	أ.سالم ف	قاعة 10	35
238	IT212	Data structures and algorithms	F	3	أ.سالم ف	قاعه 9	40
239	IT212	Data structures and algorithms	G	3	أ.سالم ف	قاعة 10	35
240	IT212	Data structures and algorithms	G	3	أ.سالم ف	قاعه 9	40

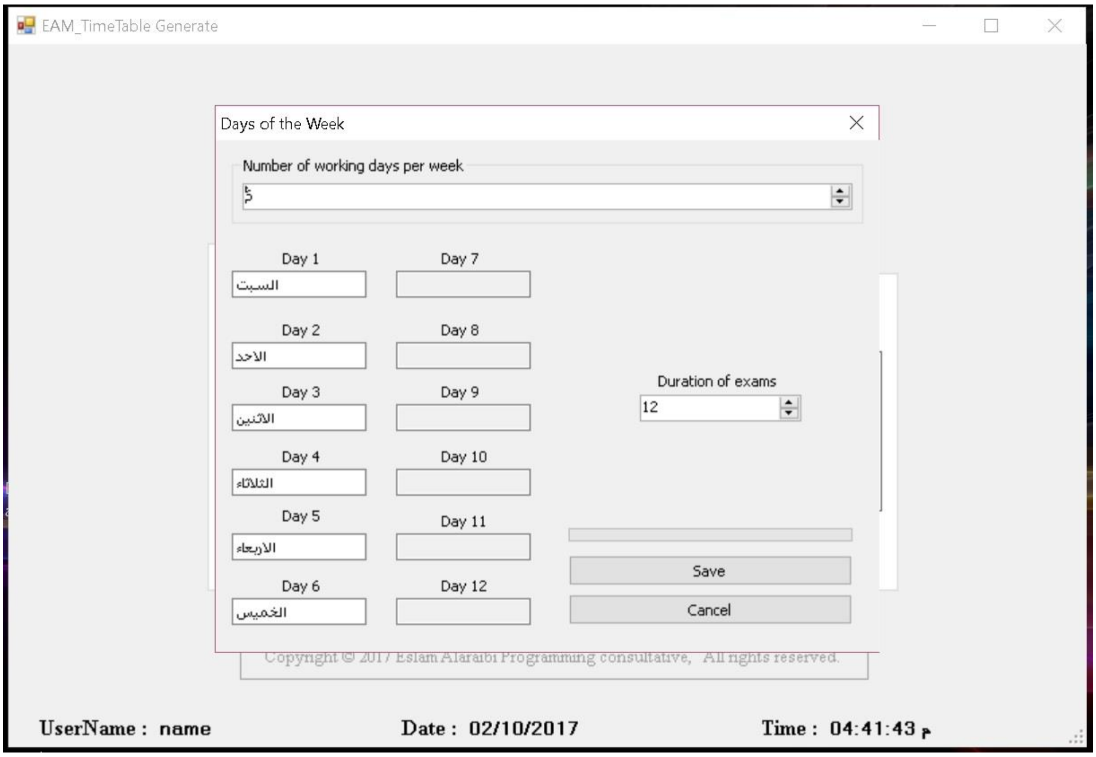
Query1							
SN	Cours_Number	ECoursName	GroupName	Credits	TeacherName	RoomName	Capacity
241	IT212	Data structures and algorithms	H	3	أ.الفرجاني	قاعة 106	50
242	IT212	Data structures and algorithms	H	3	أ.الفرجاني	قاعة 106	50
243	IT212	Data structures and algorithms	I	3	أ. القلالي	قاعة 106	50
244	IT212	Data structures and algorithms	I	3	أ. القلالي	قاعة 106	50
245	IT212	Data structures and algorithms	J	3	أ. الكيلانى	قاعه 9	40
246	IT212	Data structures and algorithms	J	3	أ. الكيلانى	قاعة 10	35
247	IT212	Data structures and algorithms	K	3	أ. الكيلانى	قاعة 316	40
248	IT212	Data structures and algorithms	K	3	أ. الكيلانى	قاعة 316	40
249	IT222	Computer Architecure	A	3	أ. معمر	قاعة 316	40
250	IT222	Computer Architecure	A	3	أ. معمر	قاعة 316	40
251	IT222	Computer Architecure	B	3	أ. معمر	قاعه 9	40
252	IT222	Computer Architecure	B	3	أ. معمر	قاعة 10	35
253	IT222	Computer Architecure	C	3	أ.فوزى	قاعه 9	40
254	IT222	Computer Architecure	C	3	أ.فوزى	قاعة 10	35
255	IT222	Computer Architecure	D	3	أ.فوزى	قاعة 316	40
256	IT222	Computer Architecure	D	3	أ.فوزى	قاعة 316	40
257	IT222	Computer Architecure	E	3	أ.فوزى	قاعة 316	40
258	IT222	Computer Architecure	E	3	أ.فوزى	قاعة 316	40
259	IT222	Computer Architecure	F	3	أ.عزالدين	قاعه 9	40
260	IT222	Computer Architecure	F	3	أ.عزالدين	قاعة 10	35
261	IT222	Computer Architecure	G	3	أ.عزالدين	قاعه 9	40
262	IT222	Computer Architecure	G	3	أ.عزالدين	قاعة 10	35
263	IT271	Network Fundamentals	A	3	أ. الراشدي	قاعة 316	40
264	IT271	Network Fundamentals	A	3	أ. الراشدي	قاعة 316	40
265	IT271	Network Fundamentals	B	3	أ. الراشدي	قاعة 316	40
266	IT271	Network Fundamentals	B	3	أ. الراشدي	قاعة 316	40
267	IT271	Network Fundamentals	C	3	أ. الراشدي	قاعة 205	30
268	IT271	Network Fundamentals	C	3	أ. الراشدي	قاعة 208	30
269	IT271	Network Fundamentals	D	3	أ.بن صريتي	قاعة 205	30
270	IT271	Network Fundamentals	D	3	أ.بن صريتي	قاعة 208	30
271	IT271	Network Fundamentals	E	3	أ.بن صريتي	قاعة 205	30
272	IT271	Network Fundamentals	E	3	أ.بن صريتي	قاعة 208	30

APPENDIX E

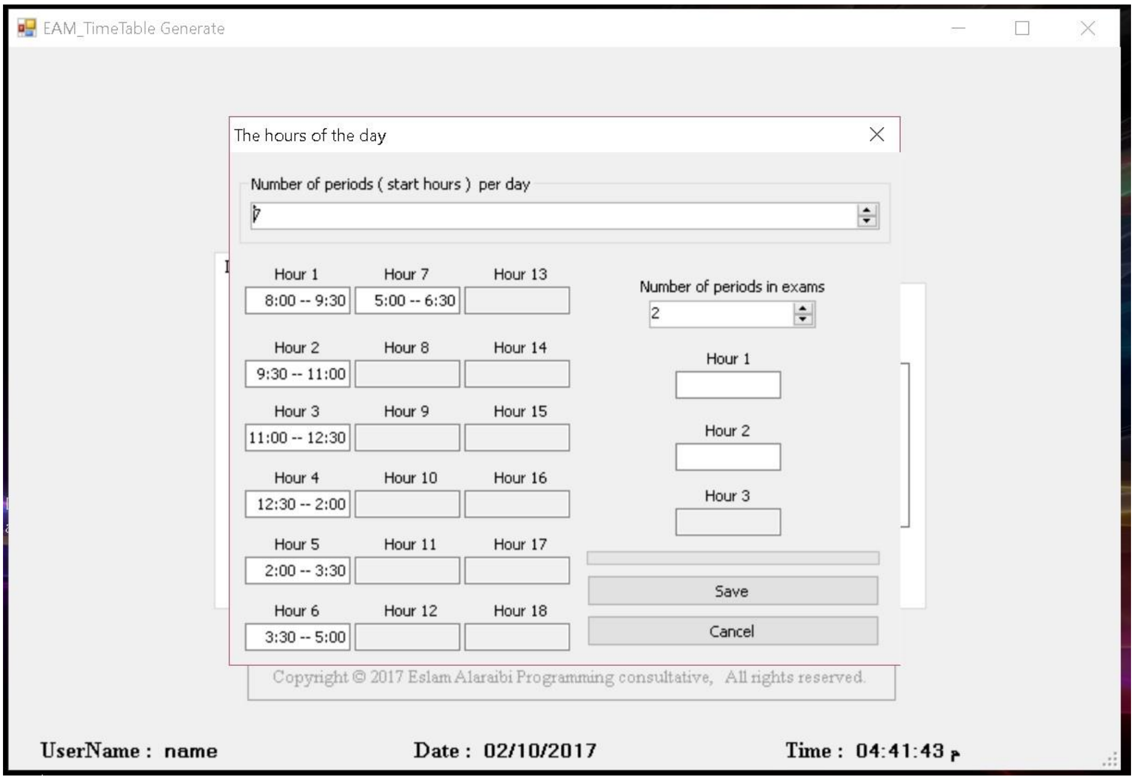
SOME SNAP SHOTS OF THE SYSTEM SCREENS



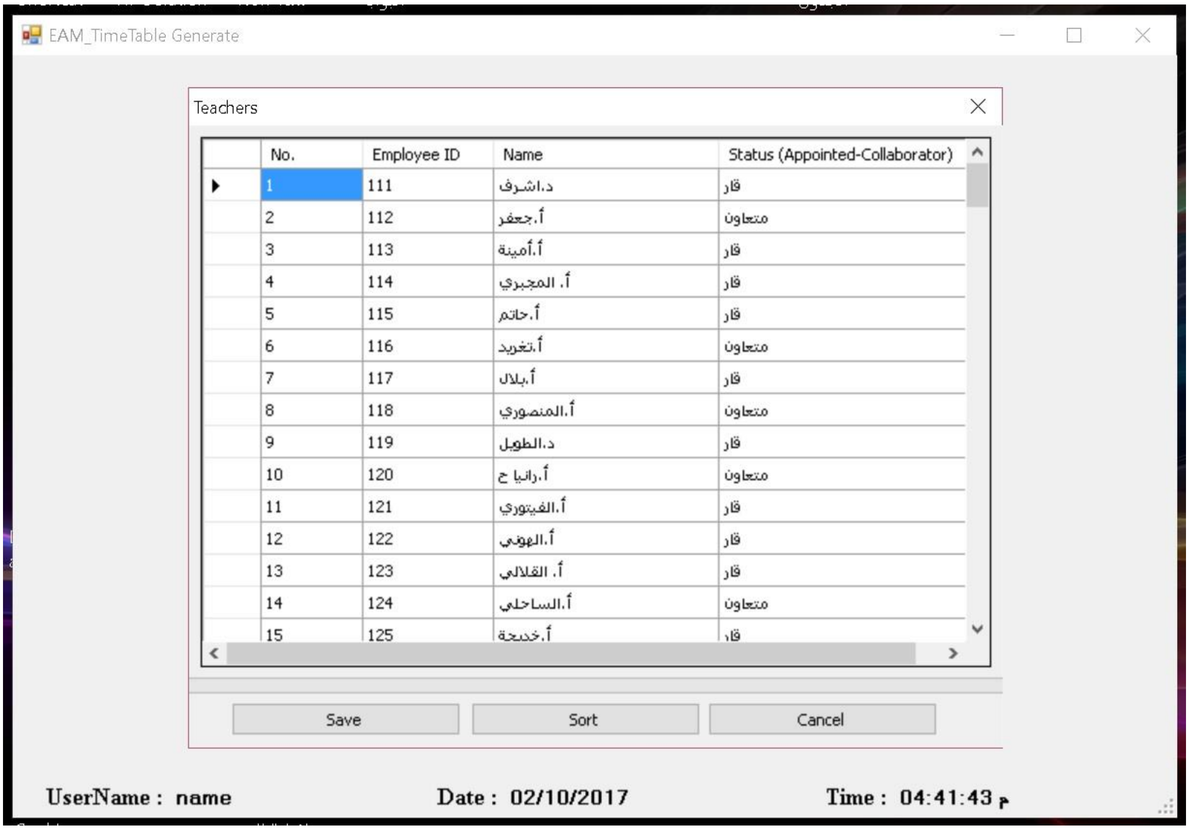
(1): Main Form



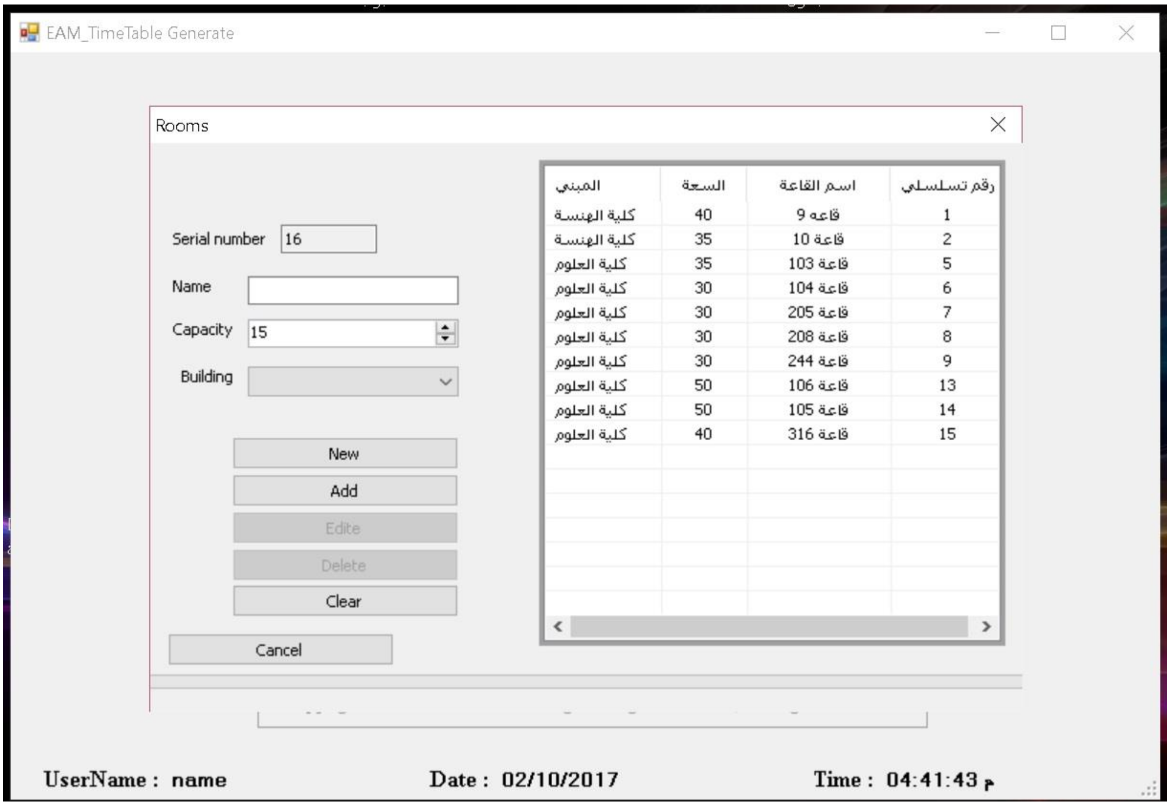
(2): Day of Week Form



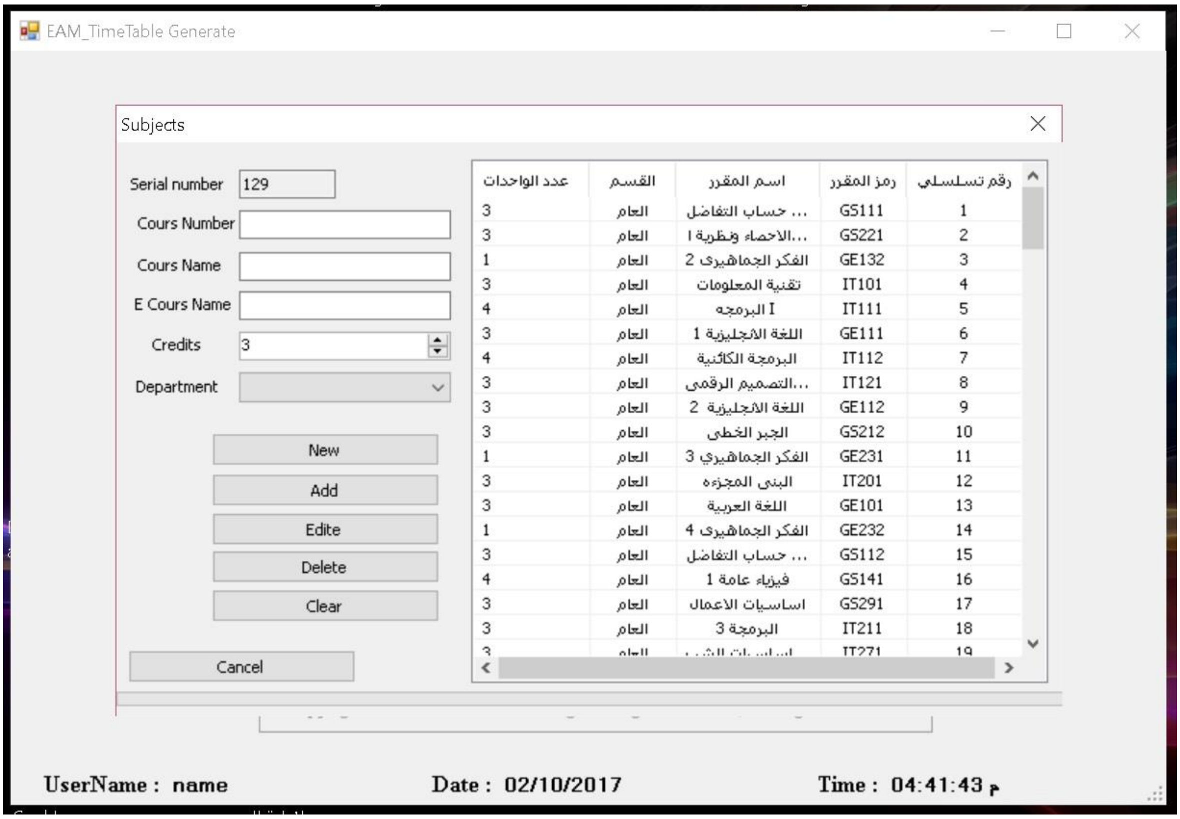
(3): Time Slots Form



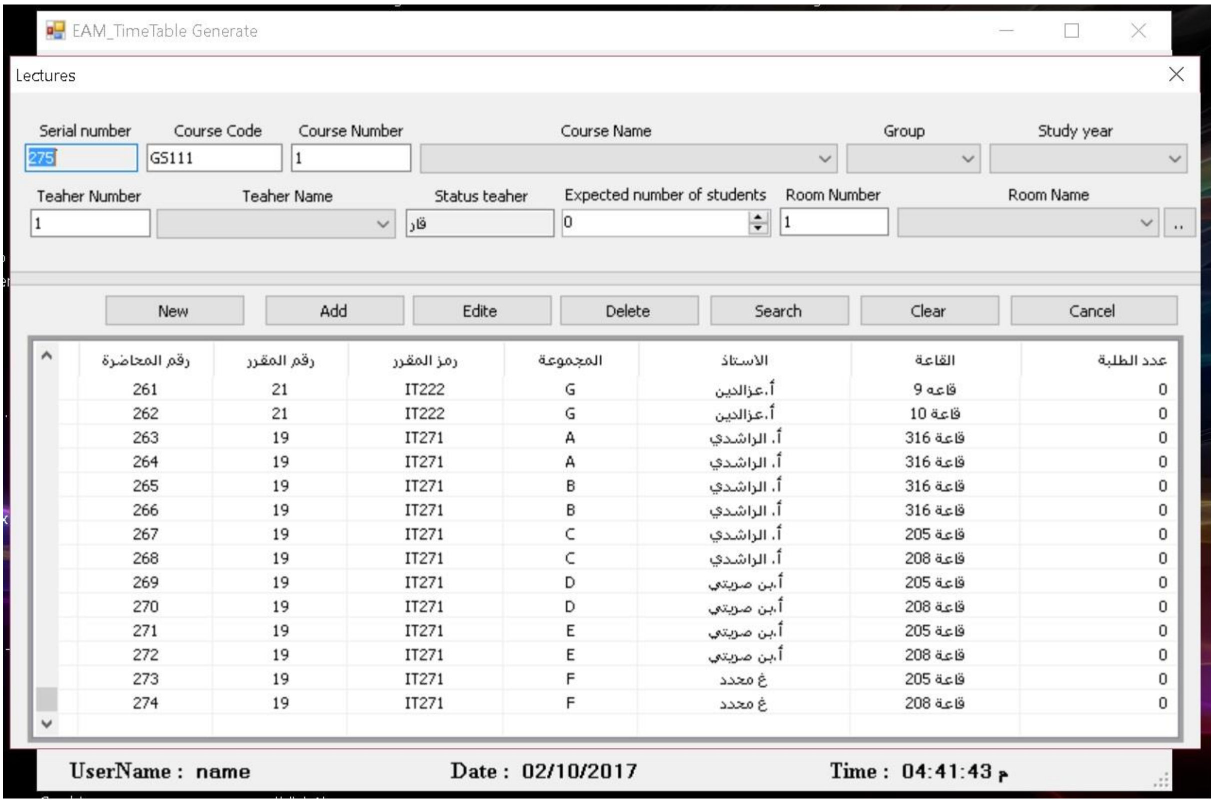
(4): Lectures Information Form



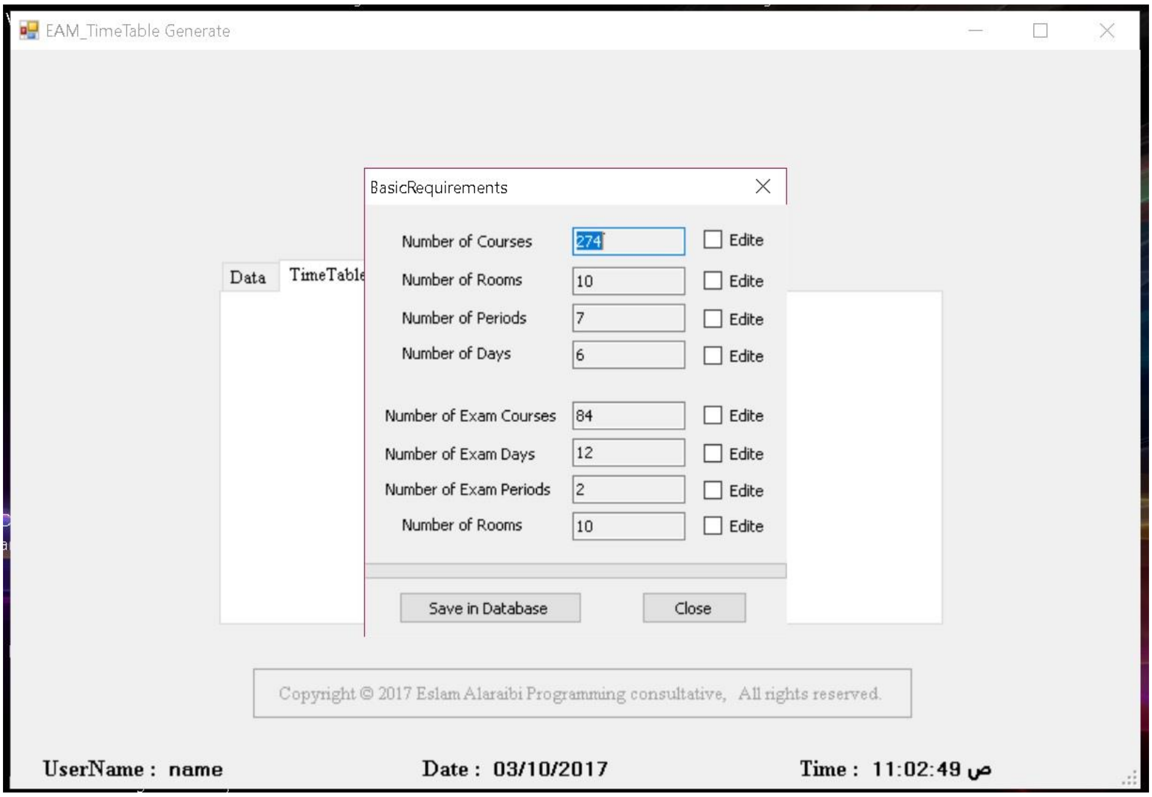
(5): Classroom Information Form



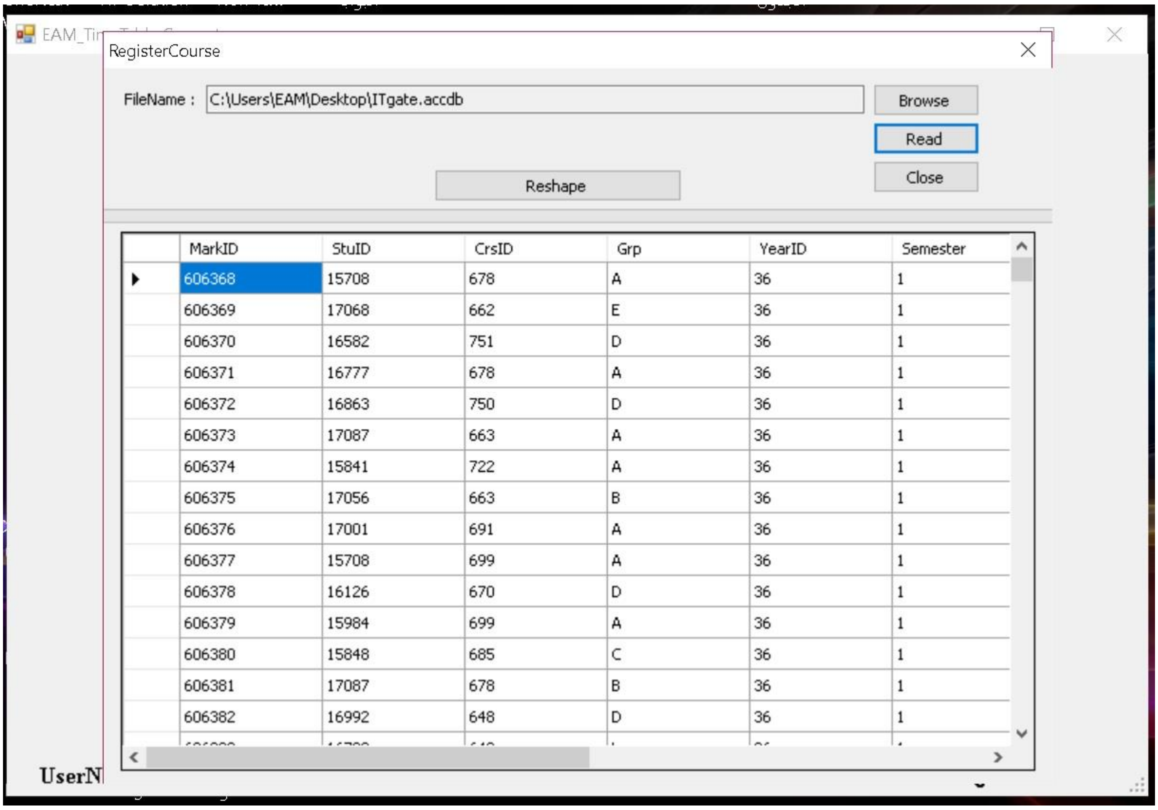
(6): Subject Information Form



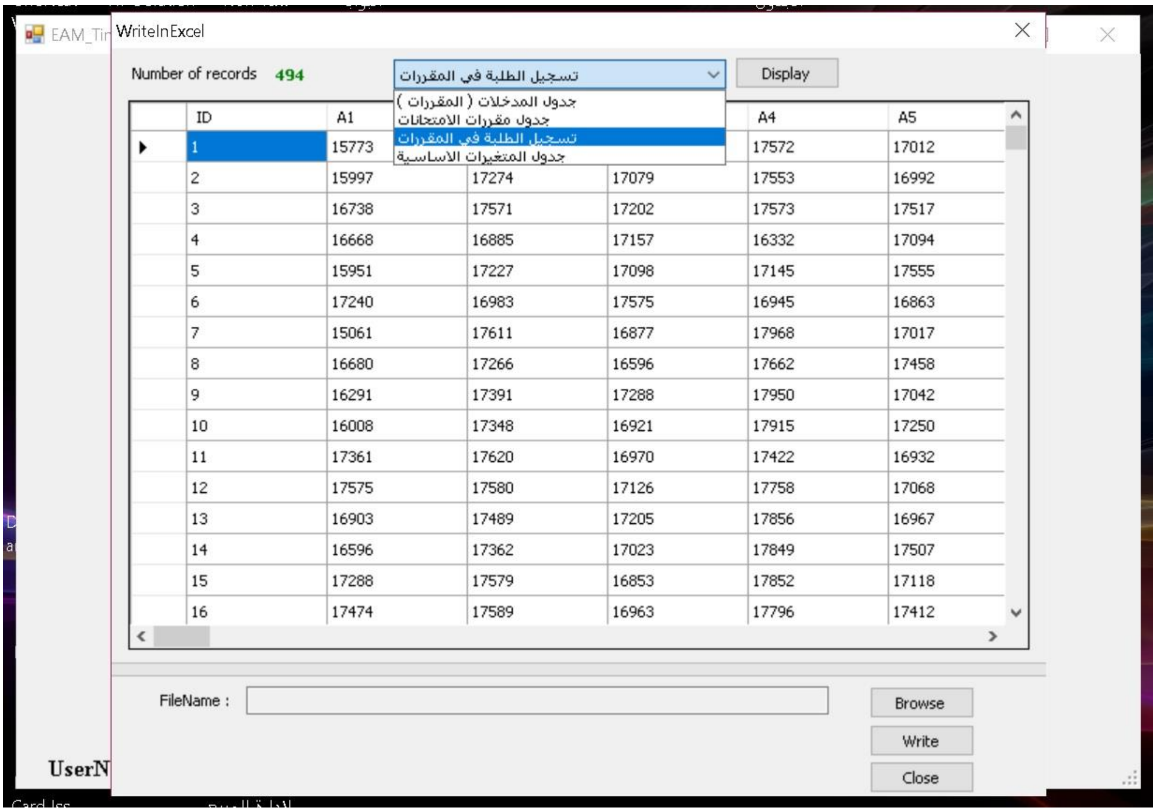
(7): Lecturer Information Form



(8): Basic Requirements Form



(9): Read Registration of Courses form



(10): Write data in Database Form

APPENDIX F

CODE EXCERPT

Initialization

```
clear all
clc
%الخوارزم الجيني الخاص باعداد الجداول الدراسي
t = cputime;
resu1=[];
xd=0;yd=0;
v=[];
p_size =2;%4
chp=30;
eeg=[]; ftt=[];
sol=[];
ta=[];
setting =xlsread('Setting.xls');
no_days= setting(4);
no_rooms=  setting(2);
no_corses= setting(1) ;
no_perods= setting(3) ;
    best_fitness=[];
load tt1 A
load tt2 B
load tt3 C
whigth = xlsread('inputD.xls');
```

Generation of generations

```
for k = 1:p_size
      s= randperm(no_perods*no_rooms*no_days);
      population(k,:)=s;
  end
```

Genetic algorithm Timetable

```
[fitness]=FITNESS (population,wlr,wr,wlc,no_days,no_perods,no_rooms,no_corses);

while fto >0
    cp=cp+4;
    if cp>120
        cp=4;
        end;
n1=10;
population=  fliplr(population);
[new_popc] =
CROSSOVER_exp3(population,n1,wlr,wr,p_size,no_days,no_perods,no_rooms,cp) ;
new_popm = MUTATE_GSA(new_popc, 0.0000001,no_days,no_perods,no_rooms);
 [fitness]=FITNESS (new_popm,wlr,wr,wlc,no_days,no_perods,no_rooms,no_corses);
size(fitness);

end
```

Constraints

```
if i==j
                A(i,j)=0;
                B(i,j)=0;
                C(i,j)=0;
         else

        if whigth(i,3)== whigth(j,3)
            A(i,j)=1 ;% same lacture in the same pareod
        end


     if whigth(i,4)== whigth(j,4)
       B(i,j)=2 ;% same classroom in the same pareod
     end

       if (whigth(i,3)== whigth(j,3))&& (whigth(i,2)== whigth(j,2))
       C(i,j)=3 ;% same lacture and same corce
       end



     if (gg(ii)> no_corses) || (gg(jj)>no_corses)
                ss1=ss1+0;
            elseif wlc(gg(ii),gg(jj)) == 3
                ss1=ss1+1;
            end
end

%-----------------------------------------------

if (spec(iii)> no_corses) || (spec(jjj)>no_corses)
                ss=ss+0;


            elseif wr(spec(iii),spec(jjj)) == 2
                ss=ss+1;


             elseif wlr(spec(iii),spec(jjj)) == 1
                ss=ss+1;
            end

end
```

الـــمـلـــخـــص

تعتبر مشكلة الجدول الزمني احد مشاكل التحسين في العالم الحقيقي، لانها توصف على أنها مشاكل متعددة الحدود (NP) ويشار إليها بمشاكل "NP- الصعبة".

في هذه الأطروحة تم استخدام الخوارزميات الجينية، وهي احدى تقنيات الذكاء الاصطناعي، للعمل علي إيجاد حلول لمشكلة الجدول الزمنية، وبالتالي الحصول على جداول زمنية من دون تعارضات مع الأخذ بعين الاعتبار شروط التحسن في هذه الجداول، مما يجعل جداول الطالاب واعضاء هيئة التدريس في أفضل صورة ممكنة و في اقل وقت و بدون اي جهد او عناء اثناء عملية توليد الجداول الزمنية. ومع ذلك، يمكن الحصول على حلول متعددة (الجداول الزمنية، كما يمكنك اختيار أفضل الجدول الذي يلبي متطلبات أفضل).

كما استخدمت الأداة المقترحة في هذه الاطروحة لأتمتة الجدول الزمني الأكاديمي لكلية تقنية المعلومات، جامعة بنغازي كحالة دراسية، والتي يمكن استخدامها لتحسين واستخدام الموارد المتاحة للجدول الزمني مثل اعضاء هيئة التدريس وأماكن التدريس والفترات الزمنية وايام المتاحة للتدريس وربط هذه الموارد بالمقررات و توزيعها علي الجدول الزمني مع الاحتفاظ بشروط و قيود صحة اعداد هذه الجداول. اما من حيث التحسين، نجد ان النتائج كانت مشجعة حيث تمت مقارنة النتائج التي تم الوصول اليها مع النتائج ذات الصلة من الباحثين الآخرين، وتظهر النتائج قدرة الاداة علي توليد المزيد من الجداول الزمنية مع وقت لا يكاد يذكر في عملية اعداد الجدول الزمنية مقارنة بالطرق التقليدية و اليدوية المتبعة.

حل مشكلة الجداول الزمنية بأستخدام الخوارزميات الجينية

(الجداول الزمنية بكلية تقنية المعلومات كحالة دراسية)

قدمت من قبل :

إسلام علي موسي العريبي

تحت إشراف:

د. عمر مصطفي الصلابي

قدمت هذه الرسالة استكمالا لمتطلبات الحصول على درجة الماجستير في علوم الحاسوب.

جـــامعة بنغـــازي

كلية تقنية المعلومات

خريف 2017