

**DEVELOPMENT OF A SCHEDULING TOOL
FOR CONSTRUCTING
A MALAYSIAN SCHOOL TIMETABLE
USING GENETIC ALGORITHMS**

SALIHIN FARAG SHOEEB

**FACULTY OF COMPUTER SCIENCE & INFORMATION
TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2003

**DEVELOPMENT OF A SCHEDULING TOOL
FOR CONSTRUCTING
A MALAYSIAN SCHOOL TIMETABLE
USING GENETIC ALGORITHMS**

BY

**SALIHIN FARAG SHOEEB
(WGC00007)**

**DISSERTATION SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SOFTWARE ENGINEERING**

FACULTY OF COMPUTER SCIENCE

AND INFORMATION TECHNOLOGY

UNIVERSITY OF MALAYA

KUALA LUMPUR

December 2003

DECLARATION

I hereby declare that the project is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UM or other institutions.

University of Malaya

ABSTRACT

The school timetabling problem is essentially the construction of a timetable for each teacher and class that satisfies the teacher requirements and which does not violate the condition that no teacher or class is scheduled more than once in the same time period. It belongs to a class of scheduling problems which is highly constrained and which is known to be NP-hard and NP-complete. A feasible timetable is one which satisfies all the hard constraints. However, to obtain a good quality timetable we have to satisfy as many soft constraints as possible. A recent approach to derive a near-optimal solution is to use evolutionary or genetic algorithms. In this project, we describe in detail the school timetable problem and present the genetic algorithm employed to construct the timetable for a typical Malaysian schools.

University of Malaya

ACKNOWLEDGEMENTS

In the name of Allah, Most Gracious, Most Merciful

I would like to take this opportunity to convey my sincere thanks and deepest gratitude to my supervisor Assoc. Prof. Dr. Raja N. Ainon for her advises, comments, suggestions, help, and invaluable guidance throughout my project.

I am also indebted to Institute of postgraduate studies university of Malaya staff and Faculty of Computer Science & Information Technology staff and management members for their technical support, suggestions, and helpful. Their insights were priceless.

I am greatly indebted to the Libyan Ministry of Education for the financial support during my study. Also, the contribution of the Libyan Embassy staff in Kuala Lumpur is highly appreciated.

Special thanks to my friends Mr. Nidal Bashir Eshah, Dr. Mohamed Elhbaibe, and Dr. Hameida Sayoud for their unlimited helpful and support.

I am very Thankful to Prof Edmund K.Burke Professor of Computer Science Automated Scheduling, Optimization and Planning Research Group School of Computer Science and Information Technology University of Nottingham UK who kept helping me understanding timetabling problem and genetic algorithms technique I used it to solve this problem.

Table of Content

Declaration	II
Abstract	III
Acknowledgements	IV
List Of Figures	X
List Of Tables	XII
1. Chapter 1: Introduction	1
1.1 Genetic Algorithms	1
1.1.1 Ga's Process And Operators	1
1.2 Objective of GA's	5
1.3 Genetic Algorithms' Applications	5
1.4 Timetabling Problem	6
1.5 School Timetable Problem	7
1.6 Dissertation Scope And Objective	7
1.7 Dissertation Methodology	8
1.8 Dissertation Organization	9
2. Chapter 2: Literature Review	11
2.1 Introduction	11
2.2 Scheduling And Timetabling Approaches	11
2.3 Timetable Problem	12
2.3.1 Educational Timetabling	13
2.3.1.1 School Timetabling	14
2.3.1.2 University/Course Timetabling	14
2.3.1.3 Examination Timetabling	15
2.3.2 Differently From The Course Timetabling Problem	15
2.3.3 Timetabling Constraints	16
2.3.3.1 Hard Constraints	16
2.3.3.2 Soft Constraints	16
2.4 Techniques Applied To The Timetabling Problem	17
2.4.1 Graph Coloring	18
2.4.2 Tabu Search	20

2.4.2.1 Tabu Search Applications	21
2.4.2.2 Tabu Search Feature and Drawbacks	21
2.4.3 Simulated Annealing	22
2.4.3.1 Simulated Annealing General Conceptual	23
2.4.3.2 Simulated Annealing Process And Operators	24
2.5 Genetic Algorithms	25
2.5.1 Biological Terminology	25
2.5.2 Current genetic algorithm practice	25
2.5.3 Encoding Of Chromosomes	26
2.5.4 Encoding Approach	26
2.5.5 Population Size	27
2.5.6 Evaluating A Chromosome	28
2.5.7 Initializing A Population	28
2.5.8 Reproduction	28
2.5.8.1 Methods Of Selection	28
2.5.8.2 Selection Conceptual	29
2.5.8.3 Crossover	30
2.5.8.4 Mutation	32
2.5.8.5 Repair Strategies.	32
2.6 Difference Between GA And Traditional Methods	33
2.7 Comparative study to GAs and other Timetabling approaches	34
2.7.1 Why Gas	34
2.7.2 Timetabling approaches comparison	35
2.8 Literature Review conclusion	38
3. Chapter 3: Methodology and Result of Survey.doc	39
3.1 Result of Survey	39
3.1.1 Introduction	39
3.1.2 Analysis Technique	40
3.1.3 Nature of the problem	40
3.1.4 Current System	42
3.1.5 Qualities of a Good Timetable	44
3.2 Methodology	46
3.2.1 Introduction	46

3.2.2 Problem Description	46
3.2.2.1 School Timetabling Constraints	46
3.2.2.1.1 Hard Constraints	46
3.2.2.1.2 Soft Constraints	46
3.2.2.1.3 Constraint Data	47
3.2.2.2 Class Clash Problem	47
3.2.2.3 Teacher Clash Problem	48
3.2.2.4 Parameters In The Representation	49
3.2.3 The Algorithms	50
3.2.3.1 Chromosome Representation	50
3.2.3.2 Initialization	52
3.2.3.2.1 Initialization Procedure	53
3.2.3.2.2 Evaluation	55
3.2.3.3 Genetic Algorithms Operators	59
3.2.3.2.1 Chromosomes Selection	59
3.2.3.2.2 Crossover	60
3.2.3.2.3 Mutation	63
4. Chapter 4: GA Design And Implementation	65
4.1 Introduction	65
4.2 Context Diagrams	65
4.2.1 Year Of Study	66
4.2.2 Teacher Information	66
4.2.3 Subject Information	66
4.3 Data Flow Diagram	68
4.4 Initialization	69
4.4.1 Calculating The Clashes	73
4.5 Genetic Algorithms Modules	73
4.6 Selection Module	74
4.7 Crossover Module	75
4.8 Mutation Module	76
4.9 The Fitness Function And Penalties	77
4.10 User interface	78
4.10.1 User data entry	78

4.10.2 GA Input/Output User Form	84
5. Chapter 5: GA Result And Discussion	96
5.1 Introduction	96
5.2 Programming Language And System Requirement	96
5.3 Data Formatting	97
5.4 Experiment	101
5.4.1 Case Study	103
5.5 Case Study Result	103
5.5.1 Test 1	103
5.5.2 Test 2	105
5.5.3 Test 3	106
5.5.4 Test 4	107
5.6 Discussion	110
5.6.1 Population Size	110
5.6.2 Generation Number	110
5.6.3 Data Quality	110
5.7 research contribution	111
5.8 MSTTP-GA Evaluation	112
5.8.1 Pilot study	112
5.8.1.1 Participants and experiment material	112
5.8.1.2 Environment	112
5.8.1.3 Methodology	113
5.8.1.4 Qualitative measurements	113
6. Conclusion And Future Work	116
6.1 Introduction	116
6.2 Conclusion	116
6.3 Future Work	117
6.3.1 GA's suggestions	117
6.3.2 Problem representation	117

Appendix A	Reference List	119
Appendix B	Experimental Materials	124
Appendix C	Questionnaire Of System Requirements	136
Appendix D	Result of Survey	145
Appendix E	Questionnaire of MSTTP-GA Tool	171

University of Malaya

LIST OF FIGURES

Figure	Description	Page
(1.1)	Genetic Algorithm Flow Diagram	2
(1.2)	Processes of Crossover	3,4
(1.3)	process of mutation	4
(2.1)	Concept of educational timetabling problem	13
(2.2)	Optimization Problem Representation Using The Black Box Model	18
(2.3)	Graph Coloring	19
(2.4)	Iteration in a Tabu search	21
(2.5)	SA algorithm Pseudo code	24
(2.6)	One point crossover	30
(2.7)	Two point crossover	31
(2.8)	Uniform crossover	32
(3.1)	Frequency of Criteria	44
(3.2)	Chromosome representation for secondary school	51
(3.3)	population of school timetable (two chromosomes)	54
(3.4)	Initial population	55
(3.5)	Offspring timetable	61-62
(3.6)	Mutation Process	63
(3.7)	Two genes mutation	63
(4.1)	Context diagram	65
(4.2)	General View of Data Flow	68
(4.3)	View Detailed of Block 2	69
(4.4)	Mapping data to chromosome	71
(4.5)	Initialize Population Algorithms	72
(4.6)	Roulette wheel parent selection	74
(4.7)	One point crossover algorithm	75
(4.8)	Two point crossover algorithms	76
(4.9)	Mutation Algorithm	77
(4.10)	Subjects Layer.	79
(4.11)	Classes Layer.	80
(4.12)	Teachers Layer.	81
(4.13)	Class/Subject Layer.	82
(4.14)	Teacher/Subject Layer.	83
(4.15)	Labs Layer.	84
(4.16)	Main UI	85

(4.17)	User reads population from a text file	86
(4.18)	Chromosomes Initialization	87
(4.19)	GA at run time	88
(4.20)	Save the best chromosome fitness	89
(4.21)	GA result	90
(4.22)	List of timetable	91
(4.23)	Master School Timetable	92
(4.24)	Master teacher timetable	93
(4.25)	Class Timetable	94
(4.26)	Teacher Timetable	95
(5.1a)	Relationship between teachers and subject	98
(5.1 b)	Subjects and their labs	99
(5.1c)	Relationship between class and number of lessons per week	99
(5.2)	Master school timetable	100
(5.3)	Number of chromosome initialized to be first members of generation	102
(5.4)	Fittest chromosome	103
(5.5)	Test 1 Result	104
(5.6)	Test 2 Result	105
(5.7)	Test 3 Result	106
(5.8)	Test 4 Result	107
(5.9)	Q1	114
(5.10)	Q2	114
(5.11)	Q3	114
(5.12)	Q4	114
(5.13)	Q5	115
(5.14)	Q6	115

LIST OF TABLES

Table	Description	Page
(2.1)	Approaches applied to Timetabling Problem	11
(3.1)	Frequencies statistics (Nature of the problem)	41
(3.2)	Descriptive Statistics Of Timetable Natural Problem	42
(3.3)	Frequencies Statistics (Current System).	43
(3.4)	Percentage of Constraint 1	45
(3.5)	Statistics of Constraint 1.	45
(3.6)	Classes Taught by Teachers	47
(3.7)	Clash relationship	48
(3.8)	Class assigned teachers.	49
(3.9)	Teacher class relationship	49
(3.10)	Example of chromosome representation secondary school	52
(3.11)	Teacher subject relationship	53
(3.12)	Class subject relationship	53
(3.13)	Constraints and values	56
(3.14)	Value distributed over periods	57
(3.15)	Value distributed over days	58
(4.1)	Lessons handled at classroom	66
(4.2)	Subject handled at lab	67
(4.3)	Relationship between subjects and classes	70
(4.4)	Relationship between subjects and Teachers	70
(4.5)	result of initialization	72
(5.1)	Malaysian secondary school	97
(5.2)	Test 1 Result	104
(5.3)	Test 2 Result	105

(5.4)	Test 3 Result	106
(5.5)	Test 4 Result	108-109
(5.6)	Simple Genetic Algorithm Results	109
(5.7)	Q1	114
(5.8)	Q2	114
(5.9)	Q3	114
(5.10)	Q4	114
(5.11)	Q5	115
(5.12)	Q6	115

University of Malaya

Chapter 1: Introduction

1.1 Genetic Algorithms

The theory of evolution proposed by Darwin in 1859 states that ‘species could be improved and could survive through the process of natural selection’. Generally, most scientific ideas are based on what is observed in nature as is the concept of Genetic Algorithms (GA’s).

GA ape the natural processes of biological evolution in real life. Based on the premise of natural selection and survival of the fittest, they provide effective solutions for optimization problems. Evolutionary computation, the first approach of genetic algorithm (GA) was developed by John H. Holland in the 1960s (Falkenauer, 1999).

(Fang, 1994) defines GA’s as:

”a groups of methods which solve problems using algorithms inspired by the processes of neo-Darwinian evolutionary theory.” (Fang, 1994)

1.1.1 GA’s Process and Operators

Encoding the target problem is normally the first step in the process of solution generation. Most regard this most difficult part in performing GA’s which are generally viewed as a ‘black box’. Just as Chromosomes contain information about the characteristics of a solution, a potential solution to a problem is defined by a set of parameters (for example teachers, classes, labs, and subject in a school timetable). These parameters are combined to form a string of values referred to as an Individual or chromosome (D. Beasley et al, 1993a).

A Genetic algorithm works as follows :

In the first population a number of chromosomes are generated randomly. Their fitness level is then evaluated. At this point the GA can start generating new populations. The following diagram illustrates every individual step of GA's (Figure 1.1).

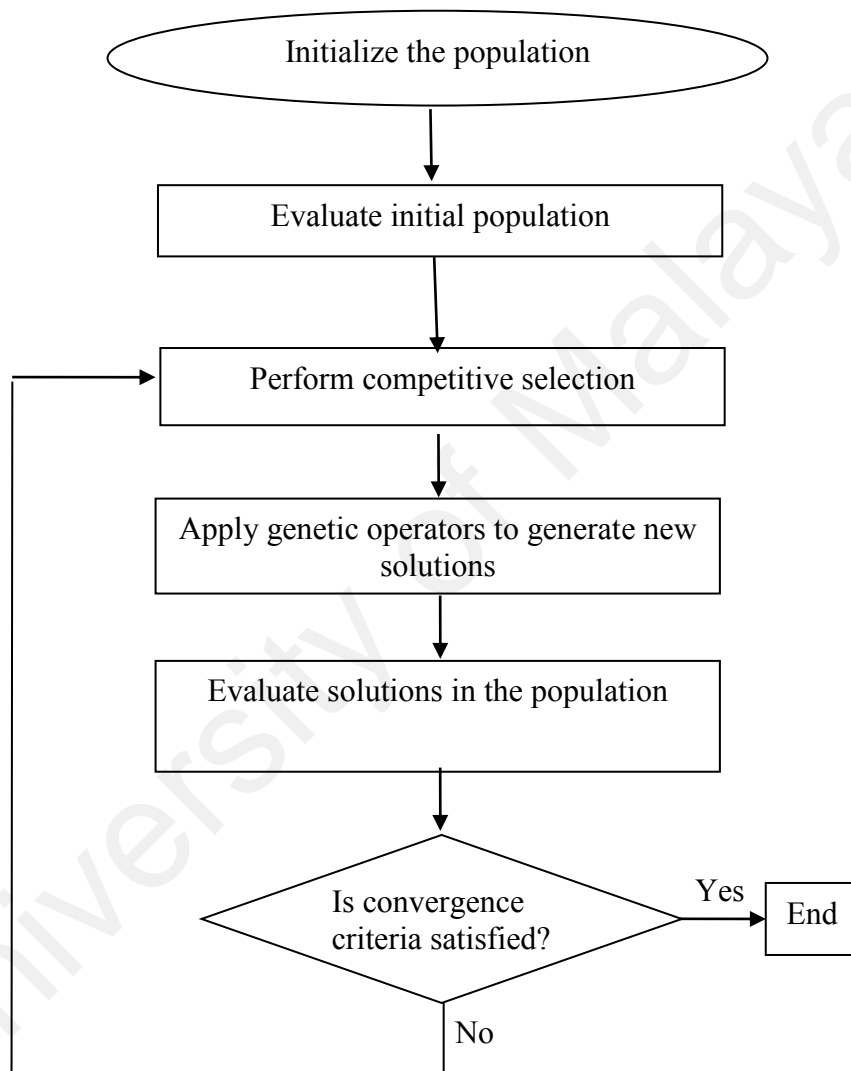


Figure 1.1: Genetic Algorithm Flow Diagram

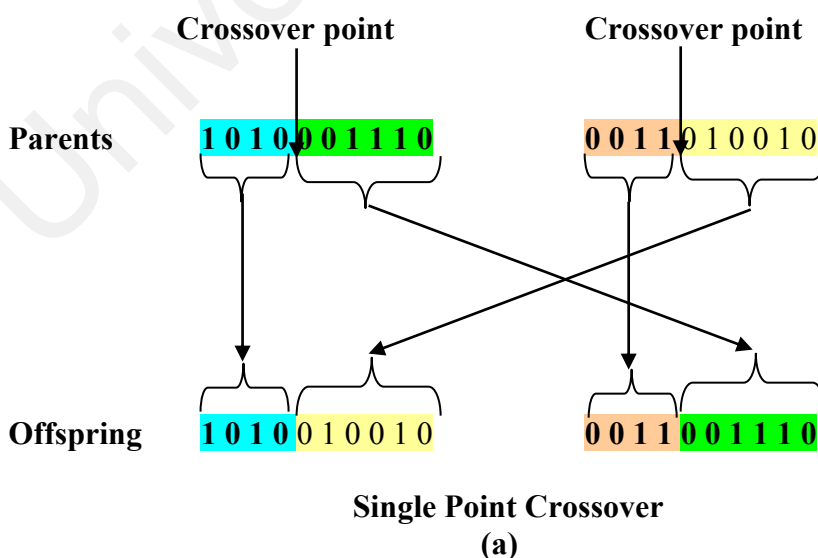
Once an initial population has been created the GA is initiated using 'Reproduction' techniques. Reproduction consists of three distinct steps:

A: Selection:

The selection method implemented is a weighted selection (Roulette Wheel, Rank Selection, Tournament Selection, Steady State Selection) where chromosomes with higher fitness levels have a better chance of being chosen. It is possible for a chromosome to be selected more than once, or even not at all. Implementation of each selection method follows a set of predefined steps.

B: Crossover:

The most important mechanism of the GA is the Crossover. Two selected chromosomes (parents) combine randomly to produce two new offspring (children). There are several variations of the Crossover. The simplest version is the one-point crossover. In this version, the chromosomes are cut (dissected) at a selected point at random to make a head and tail portion. The dissected parts are then switched around where the head of first chromosome is combined with the tail of the second one and vice versa creating two new chromosomes. The following diagram shows how this works (D. Beasley et al, 1993b) (figure 1.2).



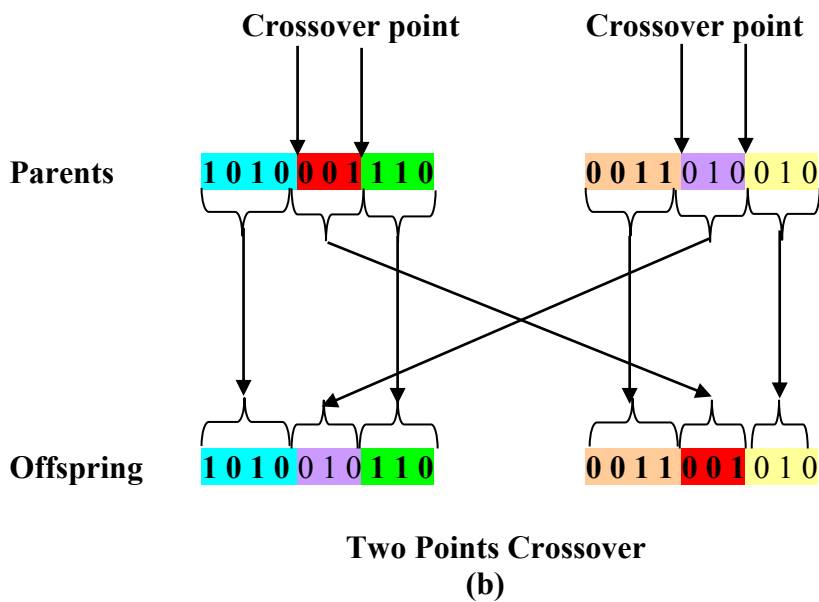


Figure 1.2: Processes of Crossover

c. Mutation:

Mutation is the next step in GA implementation. When applied individually to each chromosome, it results in some useful change. It is a mechanism which simply flips a bit in a chromosome, or if the chromosome is made out of floating point numbers, changes a number. The probability of occurrence is low (typically 0.001) (Sandikci, 2000) (figure 1.3).

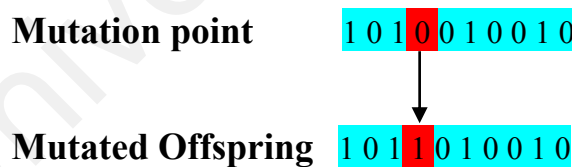


Figure 1.3: Process of Mutation

This simple process is iterated until the entire population is reproduced. The new population is then decoded and the fitness of the newly created individuals assessed. At this point, the process starts all over again. Initially it is difficult to see how this can lead to better individuals; the most important thing is that it does.

1.2 Objective of GA's

The basic **objective of GA** is to utilize the process of natural evolution in solving optimization problems. They aim to mimic nature, and in so doing, obtain an optimal or near to optimal algorithm for the computation of the optimum of a given function.

1.3 Genetic Algorithms' Application

The motivation to use GA over more traditional methods is simply due to GAs having been proven more capable of solving large complex problems where other methods have encountered difficulties. Examples are large-scale combinatorial optimization problems (such as gas pipe layouts) and real-valued parameter estimations (such as image registrations) within complex search spaces riddled with many local optima. It is this ability to tackle search spaces with many local optima which makes it popular among an increasing number of scientists and engineers. Amongst the many practical problems and areas to which GA's have been successfully applied are (David, 2001):

- Image processing
- Prediction of three dimensional protein structures
- VLSI (very large scale integration) electronic chip layouts
- Laser technology
- Medicine
- Spacecraft trajectories
- Analysis of time series
- Solid-state physics
- Liquid crystals

- Robotics Design
- Water networks
- Evolving cellular automaton rules
- The architectural aspects of building design
- The automatic evolution of computer software
- Aesthetics
- Job shop scheduling
- Scheduling
- Telecom network design
- Logistics

1.4 Timetabling Problem

The goal of Timetabling is the assignment of classes to rooms and timeslots, respecting hard constraints and taking into consideration soft constraints.

“all timetabling problems involve a set of ‘events’, things (like exams, lectures, meetings, ...etc) which need to be given a time, and a set of possible times to choose from. Sometimes there is also a choice of where events may take place, and so the problem also involves making this choice, for each event, from a set of available rooms” (D. Corne E et al, 1996).

The Timetabling problem is a form of scheduling problem. It involves processes which occur in almost all areas of our daily life, such as daily timetabling, school timetabling, or activities timetabling. Computer-based timetabling methods however, concern themselves more with simply finding the shortest timetable that satisfies all the constraints. (Fang, 1994).

1.5 School Timetable Problem

The School timetable problem is weekly scheduling for all the classes of a school avoiding teachers and groups of student double booking (Reis et al, 2000)

(Schaert,1995) describes the school timetable problem as a *class/teacher model*. He begins by describing a simplified version which can be solved in polynomial time; then he moves to the basic formulation. It has the minimal set of constraints that makes it a difficult problem, and its solution requires heuristic techniques which are employed for more complex cases.

“It is difficult to make a clear-cut distinction between acceptable and not acceptable timetables. Because of the large diversity in acceptance criteria, realistic timetable construction problems are multidimensional. Each dimension may introduce its own characteristic aspects that add to the complexity of the problem.” (Willemen, 2002).

1.6 Dissertation Scope and Objective

This dissertation explains usage of Genetic Algorithms (GAs) for finding optimal solutions to the problem of Malaysian schools Timetabling. There are two objectives in this.

1- To develop a scheduling tool to construct timetables for Malaysian schools using GA. This dissertation intends to address the Malaysian schools timetabling problem via Genetic Algorithm.

2- To obtain good quality timetables which satisfy as many soft constraints as possible. The scheduling tool process is to schedule the classes optimally such that all required constraints and much of desired constraints are satisfied. Making a distinction between hard constraints which must not be violated and soft constraints, or preferences, which it would be pleasing to satisfy but which can be violated if necessary, is one of the great difficulties with conventional approaches to timetabling. Soft constraints are constraints which may be broken, but of which breaches must be minimized. For example, More than one member of staff might need to be assigned to a particular class. Soft constraints mentioned in various papers show drastic differences. Also their order of importance appears to be a source of contention.

The scope of this project is to investigate the use of different GA operators in course timetable scheduling in Secondary schools in Malaysia. We will begin with a formal definition of our timetable construction problem. This will be followed by complexity analysis of timetable construction. The focus will be on the core problems of timetable construction, i.e., time slot, teachers, and subject group assignment. Then, a search algorithm for timetable construction problems will be developed in order to construct optimal or near to optimal solutions for real-life cases.

1.7 Dissertation Methodology

This project focuses on building a scheduling tool for Malaysian schools. Our strategy in achieving this task involves the following steps

1- Survey and analysis.

Comprehensive information and requirements about the Malaysian school timetable were gathered from number of schools. The basic collected information integrated the prospect and preferences of the timetable by school management.

Data collected from different Malaysian secondary schools analysis and compare to each others to summarize the most desired requirements.

2- Formulation of a timetabling problem for Malaysian schools using GA.

Usual framework of a genetic algorithm is used as approach to develop Malaysian school timetable tool. Our algorithm works on a population of individuals, applying crossover (one-point or two-point) and a simple mutation operator to them. As for the selection scheme, we chose roulette wheel or rank based selection method.

3- Implementing the scheduling tool as a GA.

In this project a VB program was developed which employed GA methods to perform Automated Timetabling. Consequently, the program was entitled MSTTP-GA. The GA operates upon a population of timetables which are maintained in memory. Each timetable is evaluated by testing the number of times it breaches each constraint.

4- Tests on the GA scheduling tool.

Measuring the success of our tool by conducting a pilot study using some users. This is going to be accomplished by using number of schools. The tool is going to be evaluated by collecting the users' feedback on the tool by using questionnaire.

1.8 Dissertation organization

Chapter two is divided into three main subsections. The first subsection reviews the major optimization techniques used to develop and improve timetabling. The Second subsection gives an overview of GA and their operators. The Third subsection outlines references to works presented on timetabling developed using GA's and other techniques.

Chapter three deals with the methodology and is divided into two main sections. The "results of survey on Malaysian secondary schools" section introduces the methods that have been used to gather and collect data and other requirements to develop this system. A list of Malaysian schools, diagrams showing master timetables for schools and subject listings will be included in this chapter. The "Simple GA and school timetable" section discusses the GA methodology applied to Malaysian school timetables in general terms and highlights the steps taken to design and implement School Timetable Problem (MSTTP-GA) as well as GA's operators (selection, crossover, mutation, etc ...)

Chapter four presents the implementation of the different functions of Malaysian School Timetable Problem (MSTTP-GA). It also shows how school timetable problem (MSTTP-GA) is used in executing GA.

Chapter five describes the results recorded from our studies. Finally chapter six provides a conclusion on the approach discussed in this project and evaluates the achievements of presented approach. Some suggestions for further research are also discussed.

Chapter 2: Literature Review

2.1 Introduction

This chapter describes early and current works in school timetabling. The improvements and approaches which are currently used to develop school timetable for some schools and universities, are also discussed.

2.2 Scheduling and Timetabling Approaches

As the result of survey shows there are many techniques and approaches used to develop scheduling and timetabling problems. These approaches can be categorized into three main categories and some of them are listed below (Fang, 1994) (Table 2.1)

Table 2.1: Approaches Applied to Timetabling

Exhaustive search approach	Operation research Approach	Artificial Intelligence Approach (AI)
Graph Coloring	Mathematical programming Dynamic programming Branch and bound search Heuristic search Simulated annealing Tabu search	Constraints satisfaction problem Rule-based expert system Evolutionary algorithms

This chapter describes the most used approaches applied to the timetable and scheduling problems. For problems that require optimization solutions general search strategy, such as genetic algorithms, tabu search and simulated annealing seem to be the

most promising route. Graph coloring is one of the most useful approaches applied to solving timetabling problems (A. Mehrotra and M. Trick, 1996).

In this chapter we will present these four approaches in more details as well as Genetic Algorithms, and the methodology used to solve the Malaysian school timetable problem in this dissertation. To make these techniques understood we present a brief explanation about timetabling problem.

2.3 Timetable Problem - What is the timetable problem?

Timetable is a set of events allocating to a finite number of timeslot (period) to satisfy a set of hard and soft constraints.

“Timetabling concerns all activities with regard to making a timetable. The events are usually meetings between people at a particular location.” (Willemen, 2002)

In the 1990's a number of variants of the timetabling problem have been proposed, which differ from each other based on the type of organization involved (university or school) and distinct constraints. Manual solution may take days or weeks to solve the problem. .” (Willemen, 2002)

Most of the early techniques and approaches were based on a simulation of the human way of solving the problem. More recently, some approaches based on search techniques(Schaerf,1995),(Burke et al,1997); among others, we have simulated annealing (Abramson et al,1999) tabu search and genetic algorithms(Burke et al,1995) We consider in this dissertation a problem known as school timetabling: the weekly scheduling for all the classes of a school, avoiding teachers meeting two classes at the same time, and vice versa. Our main objective was to help the management and staff of public schools in Malaysia to organize their school timetable in an optimal fashion.

2.3.1 Educational Timetabling

(willemen, 2002) identifies educational timetabling as a sub-class of timetabling for which the events take place at educational institutions. The educational timetabling problem considered as a sequence of lectures or examinations which involves teachers and students in a prefixed timeslot to satisfy a number of hard and soft constraints of numerous types. Constraints involve teacher and room availability, and student and teacher workload. This type of scheduling problems differ from each other based on the type of events, the desired number of constraints and the type of organization involved (university or school). In Figure 2.1, we show the relationship between parameters involved in educational timetabling problem.

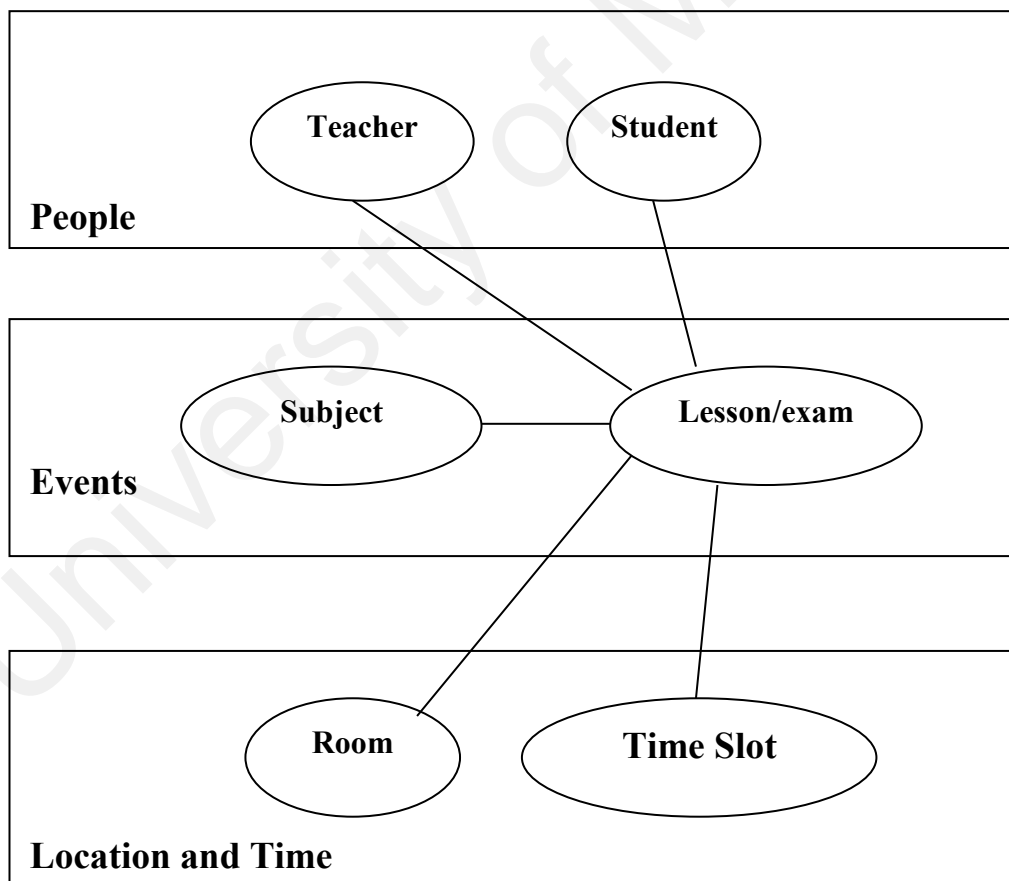


Figure 2.1: Concept of Educational Timetabling Problem

(Schaerf and Gaspero, 2001) categorizes educational timetabling problem into three main categories described in the following subsections.

2.3.1.1 School Timetabling

The school timetabling problem, known as the class-teacher problem, deals with allocating a combination of three sets (class, teacher, and subject) onto periods of the week. Therefore the problem consists in assigning lectures to periods in such a way that no teacher or class is involved in more than one lecture at a time and each teacher gives the right number of lectures to each class. Typically, each class consists of a set of students, who must be occupied from the time they arrive until the time they leave school, and a specific teacher being responsible for the class in any particular period. So the problem is to match up meetings of teachers with classes to particular time slots so that each particular teacher meets every class he/she is required to.

2.3.1.2 University/Course Timetabling

The university course timetabling problem is a weekly scheduling process for all the lectures of courses to reduce overlaps of courses having common students. (Schaerf and Gaspero, 2001) found that the difference between school and university are that the university courses could have students in common, whereas school classes are made up of disjoint sets of students having the same lessons together all the time. For example two or more of courses which share by common students cannot be scheduled at the same period. Moreover, in the school, teachers often teach more than one class, whereas in universities, a lecturer usually teaches only one course. Lastly availability of rooms and their size are very important in university timetabling problem, whereas in the school problem they are prefixed because we consider that each class has its own room

(S. Elmohamedet al, 1997), (Boyatt, 2001) , (Ross et al, 1994) , (Fang,1994), (Bambrick, 1997).

2.3.1.3 Examination Timetabling

Examination timetable simply illustrates the when and whereabouts of students exams. Some researchers consider that examination and courses timetable are similar and it is difficult to make a clear distinction between them. Nevertheless,(Schaerf and Gaspero,2001) has stated some accepted different points between examination and courses timetabling.

2.3.2 Different-from-course timetabling problem

- This kind of timetabling can be characterized as follows:
- Each subject has only one exam.
- The conflict condition is generally strict. We can accept that a student is forced to skip a lecture, but we do not accept it in exam case.
- There are different types of constraints, e.g. at most one exam per day for each student, and not too many consecutive exams for each student.
- The number of periods may vary, in contrast to course timetabling where it is fixed.
- More than one exam per room is allowed, but it is no so in lecture case.

In the universities, examination timetable is more strict and sensitive than schools, because in school situation students and room are always prefixed (students of the same class taking the same subject at the same time) but in university situation students do not share lecture all the time.

2.3.3 Timetabling Constraints

The most important part in timetabling is constraints which must be carefully specified to make sure that you are developing the right system in a clear-cut fashion (G. R. Filho and L. A. N. Lorena, 2001), (C. Blum et al, 2002), (Alberto et al, 1993), (Ainon and Yong ,1989), (Chambers,1995).

Timetabling constraints are generally categorized into two types, hard and soft constraints.

2.3.3.1 Hard Constraints

Hard constraints are those constraints that cannot be violated. In other words timetable is feasible if the hard constraints are satisfied. For a better understanding of several aspects of this see (Mukherjee,2001) Examples of hard constraints are:

- No class is assigned to more than one lesson in any particular period.
- No teacher is assigned to more than one lesson in any particular period.
- Classes must not have gaps between lessons.

2.3.3.2 Soft Constraints

(D. Corne E et al, 1996) named it as **edge constraints** instead of soft constraints. He identified it as “*things that need to be satisfied if there are to be no clashes in the timetable.*”

A good timetabling is a one that satisfies as many of the soft constraints as possible. Some examples of those soft constraints are (Voráč et al, 2001): Student and teacher do not like timetable with gaps.

- Lessons of the same subject must be distributed uniformly over the week.
- Lunch should be scheduled to specific time (an hour between 12-15).
- The number of lesson per day for teacher or class can not exceed a specified time.

Some soft constraints come on top of the priority list compared to others such as preferences involving teachers that must have higher priority than the preferences of students. Therefore the higher priority constraints have higher cost than lower ones.

2.4 Techniques Applied to the Timetabling Problem

This dissertation deals mainly with solutions to the school timetabling problem. In this section, the most used techniques to solve school timetabling problem will be discussed. Imagine that any technique (graph coloring, tabu search, genetic algorithms, ...etc) is a black box meaning that each technique has its own operators and logic flow. This is what is called optimization problem second part. Optimization problem first part represents the problem itself, such as number of parameters, the way they are presented, specifying effective factors on these problem and finally the manner to encode this problem to be ready to any technique we want to use to solve this problem (e.g., graph coloring, tabu search, genetic algorithms etc). Briefly, any optimization problem should be subjected to this meaningful analysis to be solved. The following diagram explains this idea (Figure 2.2).

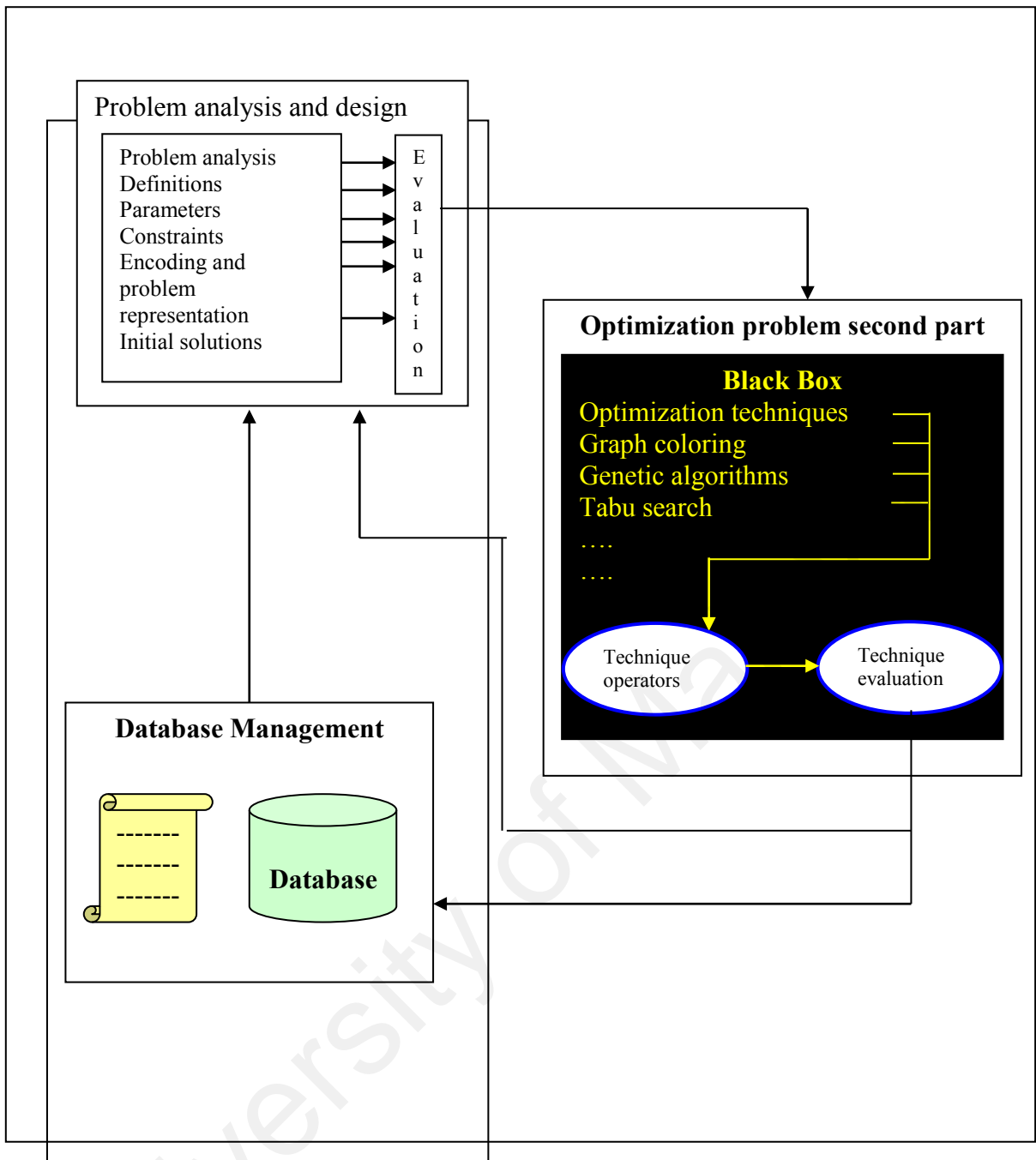


Figure 2.2: Optimization Problem Representation Using The Black Box Model.

This diagram attempts to describe the meaning of black box technique and the problem itself (optimization problem) such as scheduling and timetabling.

2.4.1 Graph Coloring

A coloring of a simple graph is the assignment of a color to each vertex of the graph so that no two adjacent vertices are assigned the same color. For example: figure 2.2

shows a graph with 5 vertices (A, B, C, D and E) and 7 edges (AB, BC, BD, BE, CD, CE AND DE) (Ainon and Yong, 1989) (figure 2.3).

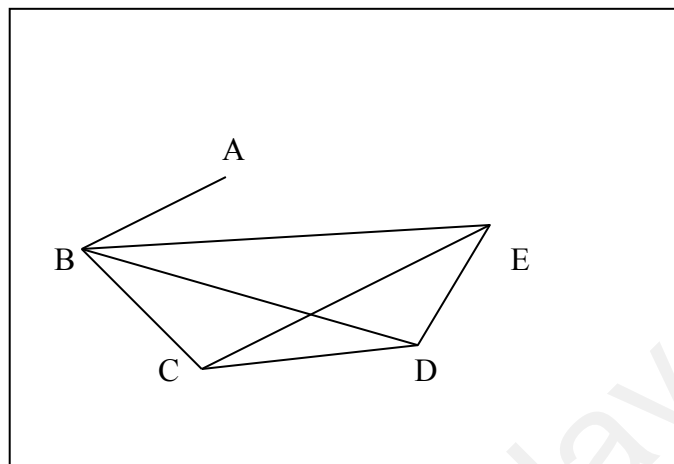


Figure 2.3: Graph Coloring.

- A and B vertices are Adjacent
- A and E Non-adjacent
- A, B, C, D and E have degrees 1, 4, 3, 3 and 3 (degree of vertex is the number of edge incident with vertex)

Researchers and system developers do not prefer using manual techniques like graph coloring in course scheduling problems. It is difficult if not impossible to model a complex problem on graph. Therefore, only simple cases can be solved by graph coloring technique (Genc, 1999)

(Burke et.al.,1994) have presented graph coloring technique to solve university timetabling problem. They assumed a graph G containing vertices x and y where $G(x,y)$ is the obtained graph. They represented each vertex by a single node connected to other nodes adjacent to x or y or both of them. Then they Combine (merge) x and y to produce x' or y' . Similarly $G(x)$ is the graph obtained by removing the vertex x and all

edges that include x from G . The Null Graph is defined to be the graph without any vertices or edges.

(Ainon and Yong, 1989) proposed graph coloring technique to solve the school timetabling problem. They present non-empty finite set of vertices and set of edges joining together. By default two joined vertices with an edge are adjacent. They presented some examples on how the vertices are adjacent or not, the degree of graph and how to calculate it.

2.4.2 Tabu Search

Tabu search is a global optimization technique proposed independently by (Glover and Laguna, 1993). This type of technique is characterized by:

- Steepest descent with memory
 - Moves through solution space
 - Uses memory techniques to avoid cycling
- A meta-heuristic superimposed on another heuristic. The overall approach is to avoid entrainment in cycles by forbidding or penalizing moves which take the solution, in the next iteration, to points in the solution space previously visited (,).
- The Moves define your neighborhood and are one of the two most important elements in a tabu search.
- The tabu List is one of the two most important elements in a tabu search.

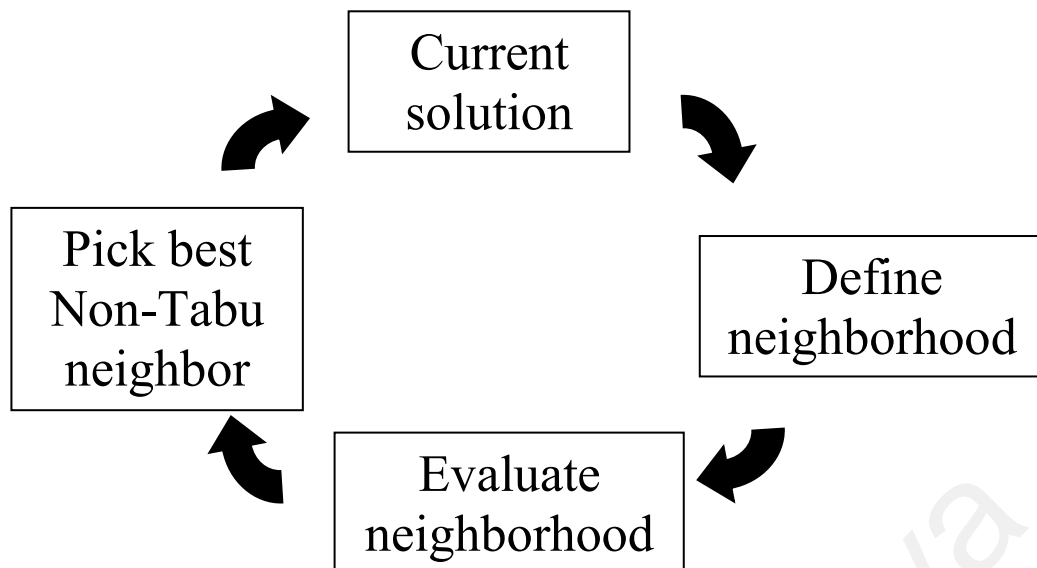


Figure 2.4: Iteration in a Tabu Search

2.4.2.1 Tabu Search Applications

Tabu search applications are quite diverse and are applied to many problems such as:

- Traveling Salesman Problem
- Knapsack Problem
- Cutting Stock Problem
- Scheduling problems
- Telecommunication path assignment

2.4.2.2. Tabu Search Features and Drawbacks

Features:

- Tabu Search yields relatively good solutions to awkward/previously intractable problems.
- Tabu Search is not bounded by linearity.

- Tabu Search provides comparable or superior solutions to other optimization techniques.

Drawbacks:

- Tabu Search does not guarantee optimality.
- Tabu Search is awkward for problems with continuous variables.
- Tabu Search assumes fast performance evaluation.
- The construction of tabu list is heuristic.

Tabu search is an iterative heuristic approach used in large sized course scheduling problems. Soft and hard constraints are required and the schedule is said to be feasible if the essential requirements and cost functions are satisfied. The aim is to find a feasible solution with a minimum cost function.

(A. Schaerf, 1996) has presented a tabu search (TS) based algorithm for the high school timetabling problem. The algorithm has produced good results for schools of various types and for different settings of the weights for the objective functions. For all cases, the timetable product turned out to be better than the handmade ones. (A. Schaerf, 1996) had applied experiments on different schools of different types and sizes to ensure that the algorithms are general enough.

2.4.3 Simulated Annealing

Simulated annealing is a modern heuristic technique for Combinatorial Problems. It is a probabilistic local search technique for finding solutions to optimization problems proposed by Kirkpatrick Gelatt and Vecchi.

2.4.3.1 Simulated Annealing General Concept

Simulated annealing (SA) is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems. SA concept is based on the manner in which liquids freeze or metals (re-crystalize) in the process of annealing. In an annealing process, a melt, initially at high temperature and disordered, is slowly cooled so that the system at any time is approximately in thermodynamic equilibrium. As cooling proceeds, the system becomes more ordered and approaches a "frozen" ground state at $T=0$. Hence the process can be thought of as an adiabatic approach to the lowest energy state. If the initial temperature of the system is too low or cooling is done insufficiently slowly, the system may become quenched forming defects or freezing out in metastable states (ie. trapped in a local minimum energy state). Simulated annealing has been used in various combinatorial optimization problems and has been particularly successful in circuit design problems.

The technique starts by creating a random initial solution. The main procedure consists of a loop that generates, at random at each iteration, a neighbor of the current solution. Like the tabu search, the definition of neighbor depends on the specific structure of the problem. Simulated annealing and tabu search are related to neighborhood family (Schaerf, 1995).

SA is a robust technique; it achieves optimization without previous knowledge of the structure of problem or solution strategy (Abramson et al, 1999).

2.4.3.2 Simulated Annealing Process and Operators

(Melicio et al, 1999) categorized the SA problem into five major classes.

1. Search Space
2. Move Set
3. Cost function
4. Annealing Scheduling
5. Data Structures

SA algorithm Pseudo code is presented below (see figure 2.5).

```
Start with the system in a known configuration, at known energy E
T = temperature = hot; frozen = false;
while (! frozen) {
    repeat {
        Perturb system slightly (e.g., move a particle)
        Compute .E , change in energy due to perturbation
        if ( $\Delta E < 0$ )
            then accept this perturbation, this is the new system config
        else accept maybe, with probability =  $\exp(-\Delta E/KT)$ 
    } until (the system is in thermal equilibrium at this T)
    If ( $\Delta E$  still decreasing over the last few temperatures)
        then  $T = 0.9 T$  // cool the temperature; do more perturbations
    else frozen = true
}
return (final configuration as low-energy solution)
```

Figure 2.5: SA algorithm Pseudo code

2.5 Genetic Algorithms

2.5.1 Biological Terminology

Biological terminology section introduces the definition of the terms used in relation with genetic algorithms.

- Chromosomes – single molecules of DNA.
- Genes – functional units which approximately encode “traits.”
- Recombination/Crossover – genes are exchanged between pairs of chromosomes to create new ones.
- Mutation – individual nucleotides are incorrectly substituted for one another.
- Recombination and mutation help create genetic diversity in the population.
- “Survival of the fittest!”
- Fitness – the probability that an organism will survive long enough to reproduce.
- Genetic Advantage/Disadvantage

2.5.2 Current genetic algorithm practice

During the last years, there has been a boom in the application of GA techniques in the solution of practical problems as can be seen by the large number of application papers in major evolutionary computation conferences (Back, 1997; Fogel et al.,1998; Koza et al., 1998; Eiben et al.,1998; Banzhaf et al.,1999).This boom in GA applications is probably due to a wide dissemination of these techniques during the past decade, and also due to the existence of more powerful computers. However, the techniques employed by users have remained nearly the same for the past 15-20 years. At first this might seem odd, but a second thought reveals that theory always need some incubation time before it is put to good use. Moreover, some of the theoretical advances are not

easily transferable to a practical context. And besides all this, some of the most important theoretical work is very recent. In summary, there is a gap between GA theory and GA practice. Up to this point we have been looking at theoretical aspects of GAs. In the remainder of the chapter we shift to the practical side of GAs and review what users do when solving practical problem with GAs.

2.5.3 Encoding of Chromosomes

A certain amount of art is involved in selecting a good decoding technique when a problem is being attacked (Davis, 1991). The first step to be taken is selecting the data types. Holland encoded chromosomes as a string of binary digits. Implementing binary encoding provides simple and effective GA's, but in some cases it is difficult to do so. There are many other methods to represent a chromosome's genes, which can be more effective for solving problems (Davis, 1991).

The chromosome is encoded as a string of symbol usually binary symbols. A binary symbol is a very simple method representing maximum information with minimum number of bit. Moreover, binary chromosomes can be operated with most GA's found in GA packages (Chan, 1994).

2.5.4 Encoding Approach

The GA works with representations of solutions rather than the solutions themselves (Falkenauer, 1999). A potential solution to a problem is represented as a set of parameters (teacher, timeslots, ...etc). These parameters (genes) are joined together to perform a string of values (chromosome) (D. Beasley et al,1993b). For example a population of a number of chromosomes which encode solutions to the problem is generated. Each member of the population consists of a number of time slots that are

allocated for the examination. Each time period contains a list of the exams that are scheduled in that period and rooms assigned to these exams (Burke and Petrovic , 2002). Chromosomes in some way contain information about a solution which they represent. The most used way of encoding is a binary string. In binary encoding, every chromosome is a string of bits, 0 or 1. The chromosome then could look like this:

C1: 110110010

C2: 110111100

Each chromosome has one binary string. Each bit in this string can represent some characteristic of the solution. Or the whole string can represent a number. Encoding depends on the problem and also on the size of instance of the problem. There are many other ways of encoding. Permutation encoding, value encoding, and tree encoding are among the many other encoding systems used in GA.

Three encoding approaches described by (Chan, 1994); direct encoding, indirect encoding, and structured encoding. Direct approach represents a solution by a chromosome. Soft constraints are penalized by decreasing the chromosome fitness.(Abramson and J. Abela,1992) used direct representation for a timetabling problem with parallel GA. Representation would produce illegal solution to amend those who need special operator on space (D. Beasley et al,1993a), (D. Beasley et al,1993b).

2.5.5 Population Size

Once the form of chromosome and type of data are specified, an entire population of chromosomes is initialized. The size of this population must be chosen. Depending on the available computing techniques used, different sizes are optimal. If the population size chosen is too small, then there is no enough exploration of the global

search space, although convergence is quicker. If the population size is too large then time will be wasted by dealing with more data than is required and convergence times will become considerably larger.

2.5.6 Evaluating a Chromosome

Generally, Random populations are almost unfit. To determine which are fitter than others, each chromosome must be evaluated. In order to evaluate a chromosome, some facts must be known about the environment in which it survives. This environment is the partially encoded (or partially decoded) part of the problem.

2.5.7 Initializing a Population

The two general techniques for initializing a population are:

A: Known Solutions a population of chromosomes (all of the genetic information about all of the chromosomes in the colony) can be loaded from secondary storage. This data will then provide a starting point for the directed evolution.

B: Random more commonly the GA can start with a random population. This is a full sized population of chromosomes whose genetic make up is determined by a random process.

2.5.8 Reproduction

2.5.8.1 Methods of Selection

The selection rate allows the user to control the amount of bias towards better individuals (G. R. Harik and F. G. Lobo and Lobo, 1999)

2.5.8.2 Selection Conceptual

Selection is the mechanism to provide parents for reproduction. Parents are chosen depending on their fitness value. This ensures that parents with higher fitness have a higher probability of being chosen. Highly fit parents could be chosen multiple times whereas parents with low fitness may not be chosen at all. The most common selection methods are:

A: Roulette Wheel Selection. Each member of the population is assigned a segment of the roulette wheel proportional to its fitness value. A random number then determines where on the roulette wheel the parent will be selected from. Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have.

B: Rank Based Selection.

Parents are sorted into order based on their fitness value. Individuals are then allocated the number of offspring they will produce based on their ranking in the population.

C: Tournament Selection.

A number of individuals are selected, their fitness values are compared and the individual with the highest fitness is selected to be the parent.

D: Uniform Random Selection.

This is used in Evolution Strategies. First the population of parents is selected based on fitness. Then parents are chosen randomly with no regard to their fitness.

2.5.8.3 Crossover

The crossover probability allows the user to control the amount of recombination or mixing (G. R. Harik and F. G. Lobo, 1999). Once two parents are chosen, breeding can take place. Crossover selects genes from parent chromosomes and creates a new offspring.

A: One Point Crossover

In this version, the chromosomes are cut (dissected) at a selected point at random to make a head and tail portion. The dissected parts are then switched around where the head of first chromosome is combined with the tail of the second one and vice versa creating two new chromosomes.

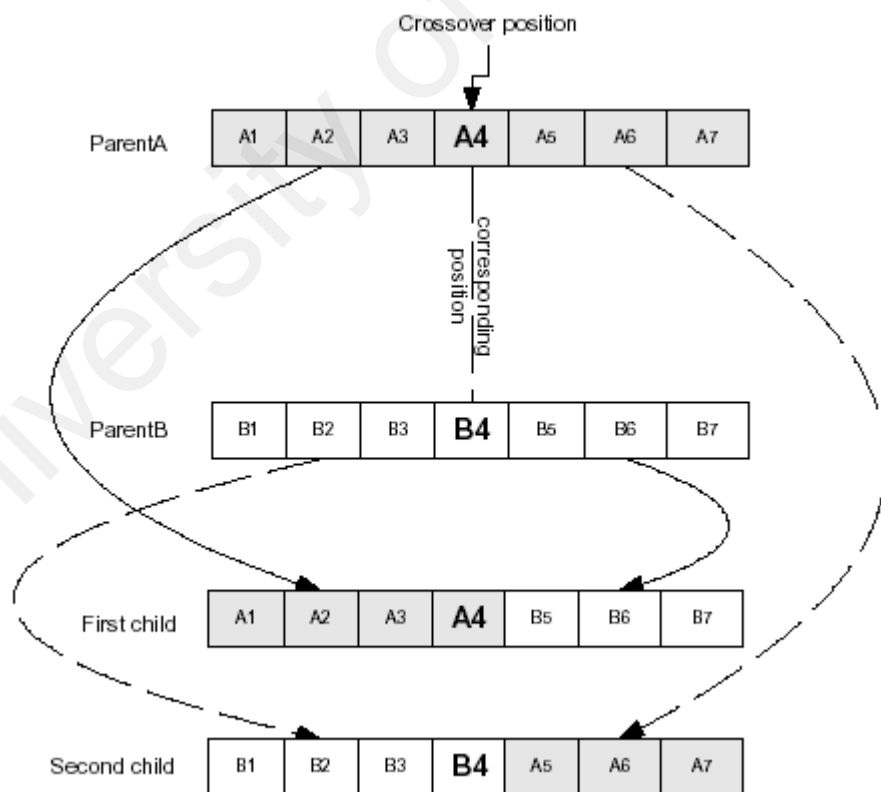


Figure 2.6: One Point Crossover (Mukherjee,2001)

B: Two Point Crossover

Two crossover points are selected where two chromosomes are cut at two random points. First and third parts of first chromosome are joined with the second chromosome sequentially and vice versa.

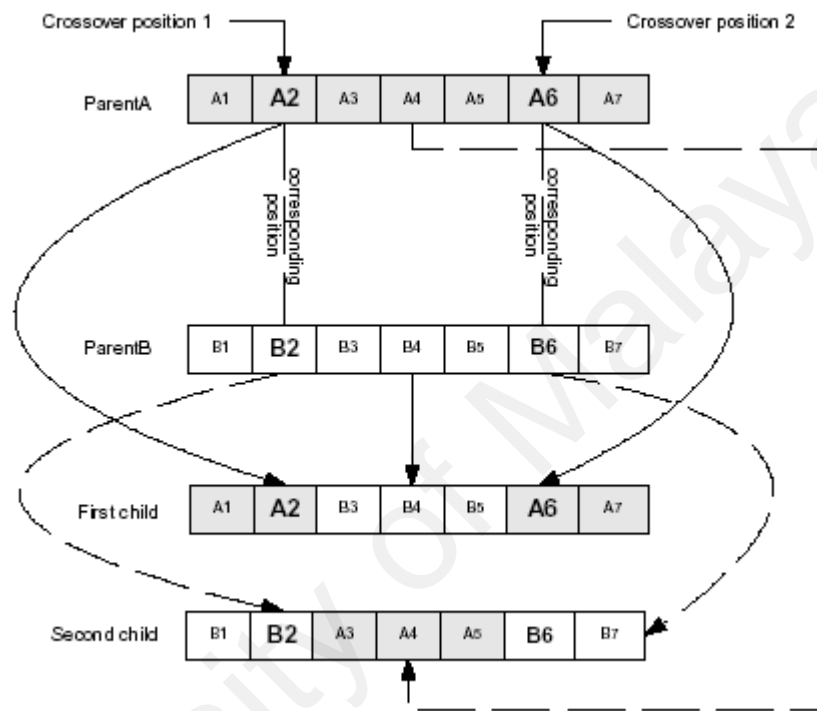


Figure 2.7: Two Point Crossover (Mukherjee,2001)

C: Uniform Crossover

Bits are randomly copied from the first or from the second parent

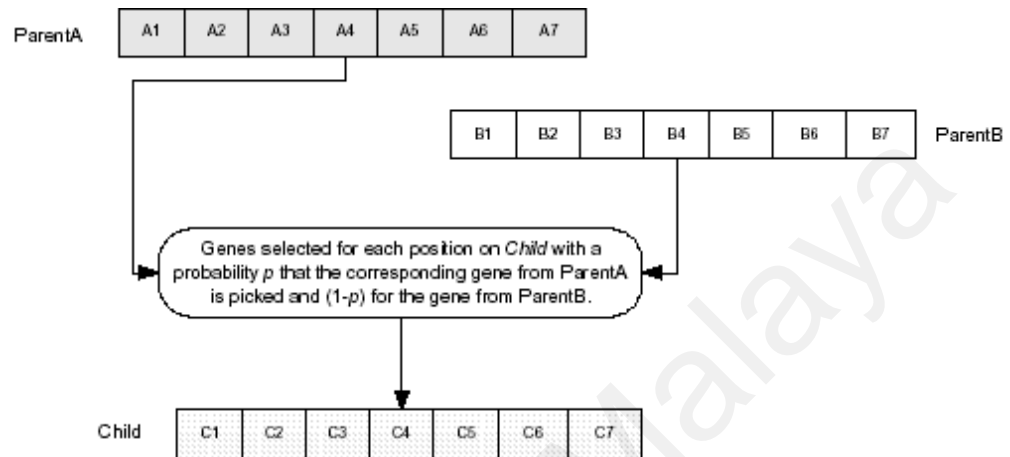


Figure 2.8: Uniform Crossover (Mukherjee,2001)

2.5.8.4 Mutation

A small change applied to the chromosome, usually replacing places of two genes with each other, is referred to as the mutation rate. The purpose of mutation is to escape local minima. Some systems do not use mutation operators at all. They rely on the noisy random populations created at initialization.

2.5.8.5 Repair Strategies.

Some representations of chromosomes can cause offspring outside the search space. In this case, a particular repair function is needed to fix that chromosome to make it more suitable for the search space.

2.6 Difference between GAs and Traditional Methods

The main differences between them are:

- Most traditional optimization methods used in science and engineering applications can be divided into two broad classes: direct search methods, requiring only the objective function values, and gradient search methods, requiring gradient information either exactly or numerically.
- Traditional methods work on point-by-point basis. They start with an initial guess and a new solution is found iteratively.
- Most of traditional methods are not guaranteed to find the global optimal solutions. The termination criterion is when the value of gradient of objective function becomes close to zero.
- GAs work with coding of the parameter set, not the parameters themselves.
- Advantage of working with a coding of variable space in Gas is that the coding discretizes the search spaces even though the function may be continuous.
- Since function values at various discrete solutions are required, a discrete or discontinuous function may be tackled using GAs.
- GAs search from a population of points, not single point so it is very likely that the expected GA solution maybe a global solution
- GAs use objective function values and not derivatives.
- In Gas, probabilistic transition rules are used, not deterministic. The search can proceed in any direction.

2.7 Comparative study to GAs and other Timetabling approaches

This section explains Gas' features and why it is needed to solve timetabling as well as advantages and disadvantages of other timetabling approaches in comparative study.

2.7.1 Why GAs

The motivation behind choosing Genetic algorithms that GAs are a general purpose optimization tool based on Darwin's theory of evolution. They have the capability to produce good quality solutions even when the dimensions of the problem increase and, for this reason, they have been successfully applied to a wide variety of problems. (Hybrid Methods for Auto. Timetabling)

Recent work in the literature shows that Genetic Algorithms have been applied to solve particular instances of the timetabling problem Genetic algorithms are search and optimization methods loosely based on the evolution in nature and were first developed by John Holland at the University of Michigan in Holland He took the idea from a book entitled The Genetic Theory of Natural Selection written by the biologist R. A. Fisher (Fisher 58) and abstracted that evolution was like learning that is a form of adapting to the environment with the difference that evolution worked over generations rather than in a single life span He thought that evolution could be seen as a creative process in essence making something out of nothing. Nowadays Gas have been used in a great number of systems to solve real life problems as explained in (Goldberg, 89 and Davis, 91) GAs have been a major breakthrough because they utilize a powerful way to perform optimization functions on a computer by improving performance over a set of options and also provide a way of studying natural phenomena There are several characteristics that clearly distinguish GAs from other techniques for solving complex problems The main one is that instead of working with a particular point in the solution

space as for example hillclimbing does GAs work with many points at the same time They use the information at a particular stage of the whole process to create a better set of potential solution points Another distinguishing feature is that they use information based on probability and random processes rather than deterministic rules to generate results The terminology used in GA has been in fact formed from terms used in natural genetics (Goldberg, 89). Strings (usually also called chromosomes) in GAs are loosely analogous to chromosomes in biological systems In GA terminology a chromosome represents an individual which is a possible solution at a particular point in the search space Chromosomes are composed of genes which describe certain features or characters and control the inheritance of one or several of those characters Genes of certain characters are in particular places in the chromosome The chromosome positions are called *loci*. The different state of values that a gene can have are called alleles The term genotype that in natural systems refers to the total genetic package in an individual is also borrowed to represent a potential solution to a particular problem In GAs the total genetic package of strings is called a structure In natural systems the phenotype is the organism formed by the interaction of the total genetic package with its environment Similarly in GAs the chromosome decodes to form a particular solution alternative or point in the solution space.

2.7.2 Timetabling approaches comparison

- To specify a GA, TS or a SA algorithm there is may well need to specify a zillion parameters so there is need to be careful to state what exactly are comparing when comparing optimization algorithms.
- Depending on what propose to call a GA TS or SA might well be a variety of other.

- Genetic Algorithms

There are some notable differences between GAs and other used techniques used to solve timetabling problem.

First, GAs work with a population of solutions simultaneously and not just a single individual solution at time. This property of GA to maintain a pool of solutions at all times makes it more robust against problems of settling in local optima.

Second, GAs are randomized approximate algorithms with rules governed by some probabilistic structure. As heuristics, GAs are not exact procedures to yield an absolutely certain global optima, but just approximate to at best yield only local optima, which may be already near-optimal or satisficing to the decision maker.

Third, in maintaining a pool of candidate solutions with randomized structures, Gas optimize the trade-off between exploring out new solutions and further enhancing current solutions (Burke, 1994).

Fourth, GAs use objective function values only, but not derivatives or other secondary information, compared to the traditional and classical methods, especially those using the Newton-type gradient or steepest descent methods (Goldberg, 1989). This property offers more generality, flexibility, and robustness that the classic methods may fail to provide in their very restrictive assumptions to perform their methods, especially with the derivative-based optimizations.

GA's are good where an integer encoding is natural for the problem and the variables are not highly correlated, i.e. crossover is likely to be an effective method of

searching the space. If X and Y are highly correlated, and they are on opposite end of your chromosome, chances are a GA search is not going to be very fruitful.

- Simulated Annealing

One of the potential drawbacks of using simulated annealing for hard optimization problems is that finding a good solution can often take an unacceptably long time (M.A. Saleh Elmohamed and Geoffrey Fox, 1998).

SA is much easier to configure because it is not as complex as the GA and because it has less parameters to set (cooling scheme, length of markov chain, method to modify parameters)

The main (potential) advantage that GA offers over SA and its kin is in the area of robustness. For easy problems, SA is faster and just as good. For hard problems, it's not so clear. For extremely hard problems on tricky search spaces with weird constraints, GA *should* do much better.

SA also has the problem of getting stuck in some local extremes. However, the process is dynamic due to the temperature, if let it run long enough, it might be able to jump out from the stagnation point. The difficulty of SA is to determine the schedule and corresponding initial temperature for a particular problem. The error surface with long narrow stiff valleys, SA can perform better than GA. This is equivalent to say that mutation with certain distribution is preferable than crossover with random cut to such error surfaces. However, if the area of extremes is not narrow but stiff, GA may work better than SA.

- Tabu Search

Tabu search (TS) is an iterative procedure designed for the solution of optimization problems. TS was invented by Glover and has been used to solve a wide range of hard optimization problems such as job shop scheduling, graph colouring (related), the Travelling Salesman Problem (TSP) and the capacitated arc routing problem. Strength and weakness of tabu search can be described as follow

- Tabu Search yields relatively good solutions to awkward/previously intractable problems
- Tabu Search is not bounded by linearity
- Tabu Search provides comparable or superior solutions to other optimization techniques
- Tabu Search does not guarantee optimality
- Tabu Search is awkward for problems with continuous variables
- Tabu Search assumes fast performance evaluation
- The construction of tabu list is heuristic

2.8 Literature Review conclusion

This chapter reviewed the basics of genetic algorithms, timetabling problems, and some other timetabling approaches. It started by giving a step-oriented review of them from an application point of view, and then moved to the most critical theoretical aspects of these algorithms. Finally, we looked at the current practice of Gas and explained what users generally do in order to solve a problem with a GA.

The differences between the theory and practice of GAs are large. The usage of advanced genetic algorithms that use adaptive operators are rarely used in practice.

Chapter 3: Methodology and Result of Survey

3.1 Result of Survey

3.1.1 Introduction

This section describes the result of survey on school timetable sent to management of thirty Malaysian secondary schools. The survey categorized into three sections as following:

- Nature of the problem
- Current System
- Qualities of a Good Timetable

First section asked questions about how many people, rooms, lessons, periods, and difficulties are related to the problem? Second section asked about how the problem is solved currently, whether a manual or automated system is used. Section three is the most important part. This section asked question about the qualities are required to develop a good timetable, in respect of sort of criteria a general automated timetabling system must meet as well as those which improve the quality of timetable. Those criteria defined as soft and hard constraints.

This survey is of the school timetabling Problem. Its main aim was to discover how the requirements of each school differ and whether these differences are sufficiently small that a unified system may be produced? Specifically, the survey aimed to answer the following questions. How prevalent is computer aided timetabling in Malaysian secondary schools? What functionality must a system have? How big and complex is the timetabling problem and how does this vary between schools? What properties must an acceptable timetable possess before an institution will use it? Twenty out of thirty schools replied to the survey.

3.1.2 Analysis Technique

SPSS 11.5 for Windows used to analysis data gathered by questionnaire (see Appendix C). SPSS stands for the "Statistical Package for the Social Sciences". SPSS is from SPSS, Inc. SPSS is an integrated system of computer programs to analyze social science data. SPSS is capable of :

- Input from almost any type of data file
- File management, including sorting, splitting, and aggregating files, matchmerging multiple files, and saving fully defined system files
- Data management, including sampling, selecting, and weighting cases, recoding variables, and creating new variables, using extensive numeric and string functions
- Tabulation and statistical analysis
- Report writing

3.1.3 Nature of the problem

In the survey the following questions were asked to clarify the school timetable problem:

1. How long is each period?
2. What is the maximum number of period a week?
3. How many lessons per week must each teacher teach?
4. Are there any lessons/subject, which has to combine students from different classes?
5. Are students allowed to have lessons both in the morning and afternoon?
6. Are there different break times for the different years in the school?
7. What is the maximum capacity of a classroom?

8. What is the population of students of this school?
9. What is the population of teachers of this school?

This survey is looking for the most common answers between schools to consider them as basic requirements for this system. Questions have divided into two groups which are frequencies statistics questions and descriptive statistics questions depends on the natural of question. Questions 4,5, and 6 have two probability of answer which are yes or no, so these questions considered as frequencies statistics questions (table 3.1). see appendix D part 2.

Table 3.1: Frequencies statistics (Nature of the problem)

Question 4	Frequency	Percent
Valid yes	15	100.0
Question 5		
Valid no	12	80.0
yes	3	20.0
Total	15	100.0
Question 6		
Valid no	8	53.3
yes	7	46.7
Total	15	100.0

Table 3.1 describe frequency of each question for example question 4 answered as yes answer by all the schools. This group of question is different from descriptive statistics questions where the first group can be yes or no. second group which is descriptive statistics questions have different answer in which question can represented by different value.

Table 3.2 describes descriptive statistics questions. This group of questions concentrate on the value of minimum and maximum answer. Question 1 has value 35 as minimum which mean that only few of schools assign 35 minute for individual period and most of school assign 40 minute for each period.

Table 3.2: Descriptive Statistics Of Timetable Natural Problem

	N	Minimum	Maximum
How long is each period	15	35	40
What is the maximum number of period in one week	15	40	42
How many lessons per week must each teacher teach	14	26	30
What is the maximum capacity of a classroom.	15	35	45
What is the population of students of this school	15	880	2600
What is the population of teachers of this school	15	53	135

3.1.4 Current System

In the survey questions were asked about the methods used by school to produce timetable, functionality, usability, and associate difficulties to produce school timetabling system (table 3.3).

1. Is a computer used at any stage in the timetabling process?
2. Is a new timetable generated each year?
3. What is/are the major cause(s) of change?
4. Is manual manipulation of the timetable allowed?
5. How would you rate the timetables produced by your system?
6. What is the maximum amount of time available between receiving the data and having to produce the timetable?

7. How long does it usually take to produce the timetable from receiving the data to printing the final version?
8. What are the main reasons for this change?

Table 3.3: Frequencies Statistics (Current System).

Question 1		Frequency	Percent
Valid	Yes	4	26.7
	No	11	73.3
	Total	15	100.0
Question3		Frequency	Percent
Valid	Staf	4	26.7
Missing	System	11	73.3
	Total	15	100.0
Question 4		Frequency	Percent
Valid	Yes	4	26.7
Missing	System	11	73.3
	Total	15	100.0
Question 5		Frequency	Percent
Valid	Satisfactory	10	78.1
	Good	3	13.3
	Excellent	2	8.6
	Total	15	100.0
Question 6		Frequency	Percent
Valid		11	73.3
	2Week	3	20.0
	3Week	1	6.7
	Total	15	100.0
Question 7		Frequency	Percent
Valid		11	73.3
	2Day	1	6.7
	2Week	1	6.7
	3Day	2	13.3
	Total	15	100.0
Question 8		Frequency	Percent
Valid	Late Data	2	46.8
	Subject Clash	8	53.2
	Total	15	100.0

3.1.5 Qualities of a Good Timetable

There are a vast number of possible constraints on the timetable produced. These relate directly to the quality of the timetable in terms of its usability and requirements on those who are subject to it. The survey asked about numbers of the more common requirements, ranking each in terms of its usage, its importance and how many lessons it affects. Space was also provided for schools add their own constraints.. The survey assumed that the constraint that no teacher should be timetabled in more than one period at once.

Appendix D – part 2 gives the survey results for each of constraints, these being ordered by the number of Malaysian schools that use them. The x axis gives the importance of the constraint (rated fundamental, desirable or irrelevant) and the y axis, the percentage of constraints. Figure 3.1 illustrated constraint 1 which is “no teacher is assigned to more than one lesson in any particular period.” It is show that this constraint is fundamental for all the schools.

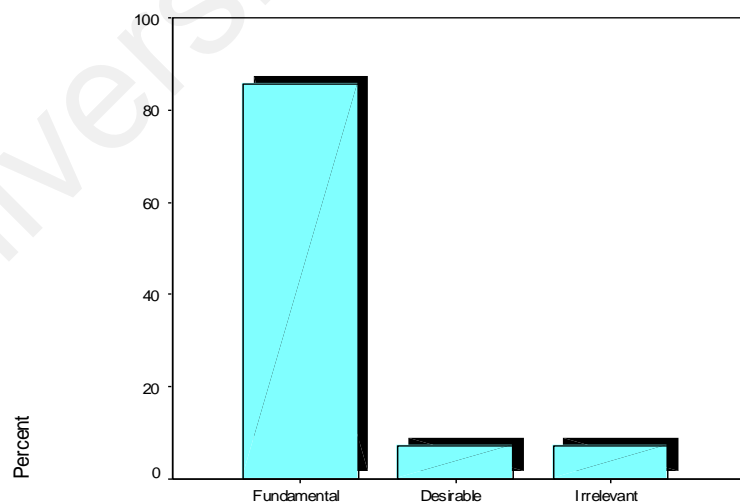


Figure 3.1: Frequency of Criteria

Table 3.4 describe constraint 1 in details first column represent criteria of constraints, second column represent the frequently criteria and third column represent percentile of the current criteria. For example: table 00 show that 12 school of 15 answered question 1 as fundamental, 1 as desirable, 1 as irrelevant, and 1 no answer. This means that this question has higher priority for school timetabling.

Table 3.4: Percentage of Constraint 1

		Frequency	Percent
Valid	Fundamental	12	80.0
	Desirable	1	6.7
	Irrelevant	1	6.7
	Total	14	93.3
Missing	System	1	6.7
Total		15	100.0

Table 3.5: Statistics of Constraint 1

Statistics

No teacher is assigned to more than one lesson in any particular period.

N	Valid	14
	Missing	1
Minimum		1
Maximum		3

Table 3.5 describe valid number of schools those answered question 1 and minimum and maximum chosen answer.

3.2 Methodology

3.2.1 Introduction

We chose to use the usual framework of a genetic algorithm for our approach. Our algorithm works on a population of individuals, applying crossover (one-point or two-point) and a simple mutation operator to them. As for the selection scheme, we chose roulette wheel or rank based selection method.

3.2.2 Problem Description

3.2.2.1 School Timetabling Constraints

There is a number of requirements that must be met to obtain a feasible timetable and a number of requirements that should be satisfied (as many as possible). These requirements are described as constraints in this dissertation and can be categorized into two categories

3.2.2.1.1 Hard Constraints

Examples of these are

- No teacher is assigned to more than one lesson in any particular period.
- Classes must not have gaps between lessons.
- The number of lessons per week for a class cannot exceed a specified limit.

Hard constraints have to be taken into consideration very strictly, because the timetables that violate just one of these are unusable

3.2.2.1.2 Soft Constraints

Examples of these are

- Science subjects must be scheduled early in the timetable.

- Some lessons may only be scheduled to particular periods.
- Teachers prefer to have more lessons on some days, in order to have a day without lessons (free day).
- Lessons of the same subject for a class must be distributed uniformly over the week.

The timetable that violates these constraints is still usable, but it is not convenient for neither, students nor teachers.

3.2.2.1.3 Constraint Data

In order to test each type of the hard constraints it is necessary to store sufficient detail about the school (i.e. this information concerning all teachers, and their subjects). The way in which each of these data types are implemented will now be given in detail.

3.2.2.2 Class Clash Problem

Obviously, teacher will teach more than one class each week. When a teacher teaches two classes, the two classes can not scheduled at the same time or a teacher clash will happen. Rephrasing, two or more teachers are clashed if they were assigned to the same class at the same time.

Table 3.6: Classes Taught by Teachers

Teachers	Class	
Rose	Class 1	Class 6
Nidal	Class 1	Class 3
Omar	Class 2	Class 7
Adil	Class 3	
Ali	Class 4	Class 8
Ahmed	Class 5	

(Table 3.6) shows an example case for classes that are taught by 6 teachers. According to our definition, Class1 and Class 6 are clashed since they are taught by the same teacher, Rose. This means that Class 1 is also clashed with class 3, taught by Nidal. The clash relationships are shown in (table 3.7), where similarly filled cells indicate that there is a clash between the two teachers.

Table 3.7: Clash Relationship

Class	Class1	Class2	Class3	Class4	Class5	Class6	Class7	Class8
Class1								
Class2								
Class3								
Class4								
Class5								
Class6								
Class7								
Class8								

3.2.2.3 Teacher Clash Problem

This is the same case of clash as teacher clash with the exception of it involves classes rather than teachers. Of course, classes taught by more than one teacher or teachers who teach the same class can not be scheduled at the same time to avoid clash occurring.

Table 3.8: Class Assigned Teachers.

Class	Teachers	
Class 1	Rose	Nidal
Class 2	Omar	
Class 3	Nidal	Adil
Class 4	Ali	
Class 5	Ahmed	
Class 6	Rose	
Class 7	Omar	
Class 8	Ali	

Nidal and Rose are clashed since they are to teach the same class, Class 1 that Nidal and Adil are clashed as well if they were scheduled to teach the same class at the same time, Class 3. Table 3.9 shows how a teacher clash can occur.

Table 3.9: Teacher Class Relationship.

Teacher	Rose	Nidal	Omar	Adil	Ali	Ahmed
Rose						
Nidal						
Omar						
Adil						
Ali						
Ahmed						

3.2.2.4 Parameters in the Representation

In this dissertation the Malaysian secondary schools timetabling problem is considered. School timetabling is a process of assigning teachers, Students, Classes, Time, and Rooms. In our problem though, we do not consider either the students or

rooms since they are fixed by the school management. Since we are not including the students and rooms directly into the representation, our problem now is assigning teacher and subjects to certain classes and time slots (period). We need to create a chromosome with three segments and we refer to time slot (period) by location of this segment. We consider a direct representation instead of the more common indirect representation.

- Teacher + subject (fixed)
- classroom + students (fixed)
- Period (time slot)
- Class

3.2.3 The Algorithms

3.2.3.1 Chromosome Representation

The scheduling problem of Malaysian secondary schools needs to be refined, to be able to adapt it to the genetic algorithms techniques. The first step is to come out with a representation for timetable solution, mapped into chromosomes. The elements that exist in the timetable such as teacher and time are coded into string codes, directly mapping them into chromosomes. In our case, the genes of the chromosome (individual) represent the teacher plus subject, and the alleles of genes are timeslots. (Figure 3.3) shows direct representation method. Timetable of Malaysian secondary schools deals with fixing a sequence of weekly meetings between teachers and students in prefixed room and time slot (period) in such way that no clashes occur. Our weekly timetable problem is 5 days, each day is divided into 9 time slots (periods), each period is 40 minutes, resulting in a total of 45 periods a week numbered from 0-44 . There are several subjects which must to be taught, each subject is divided into a number of lessons over the week. A subject can be a lesson, a lab or lesson of groups. For example two classes or more (in the same study year) can be combined and divided into two or

more (depending on number of combined classes) groups. Moreover, students of the same year with different classes (rooms) have different elective subjects. Our objective is to develop a program using simple GA that can easily be used in secondary Malaysian schools considering elective and group subjects that other tools did not consider. The program we aim to develop must be able to implement a number of constraints and maximize others (soft and hard constraints).

The timetable for a school is a matrix of $C \times P$ size as shown in Figure 3.2 There are two ways for chromosome representation in the timetable

- Fixed-length string, in this way a gene represents teacher and subject and alleles represents a timeslot.
- Fixed-length string, in this way genes are time slots and alleles are teachers and subjects (Kiang,1998)

Day	Day 1					Day 2			
	P1			P2	...	P9	P1	...	P9
Class	Subject	Teacher	Lab						
Classroom 1									
.....									
.....									
.....									
.....									
.....									
.....									
Classroom n									

Figure 3.2: Chromosome Representation for Secondary School

Each chromosome represented as a matrix of $C \times P$ size, where C represents classrooms (C_1, C_2, \dots, C_n) and P represents time slots (periods) (P_1, P_2, \dots, P_9). The

intersection point of C,P represents teacher, lesson and whether this lesson lab or not.

Table 3.10 shows real example of chromosome representing secondary school timetable.

Table 3.10: Example of Chromosome Representation Secondary School.

Day	Monday				Thursday																Friday			
Class	P1	P2	P1		P2		P3		P4		P5		P6		P7		P8		P9		P1
Class1					MA	27	MA	27	BI	5	BI	5	SNL	7	SNL	7	GE	29	KHL	13	KHL	13		
Class2					BI	6	BI	6	KHL	13	KHL	13	PS	5	PS	5	BC	16	GE	12	MA	27		
Class3					KHL	13	KHL	13	BM	25	BM	25	MA	2	SE	6	SE	6	SNL	7	SNL	7		
Class4					BM	25	BM	25	MA	2	BI	6	GE	12	KHL	13	KHL	13	PS	5	PS	5		
Class5					SE	9	PD	17	PJ	8	PJ	8	BM	32	BM	32	MA	10	MA	10	SN	28		
Class6					MA	24	MA	24	BI	11	BI	11	SNL	21	SNL	21	PD	17	PD	17	PN	16		
Class7					PJ	16	PJ	16	SNL	30	SNL	30	BM	14	BM	14	PM	24	PM	24	SE	18		
Class8					PJ	7	PJ	7	BM	32	BM	32	MA	24	SE	9	SN	28	BI	11	BI	11		
Class9					SE	18	SE	18	PM	14	PM	14	BI	11	BI	11	SNL	30	SNL	30	MA	24		
Class10					MT	10	MT	10	BI	22	BI	22	BM	23	BM	23	KML	26	KML	26	SE	9		
Class11					PM	12	PM	12	PJ	16	PJ	16	PS	19	PS	19	PS	19	EA	31	EA	31		
Class12					SNL	26	SNL	26	BM	23	BM	23	EA	15	EA	15	PN	23	PM	14	MA	30		
Class13					SN	21	SE	9	MA	28	MA	28	BM	3	BM	3	CP	25	CP	25	BI	8		
Class14					BM	3	BM	3	PA	31	PA	31	PM	29	MA	30	SN	21	BI	22	BI	22		
Class15					PA	31	PA	31	MA	10	MA	10	BI	8	BI	8	EA	15	EA	15	PM	14		
Class16					MA	28	MA	28	SNL	26	SNL	26	BI	22	BI	22	CP	9	CP	9	PD	17		
Class17					PN	29	EA	15	PD	17	PD	17	BM	18	BM	18	SE	32	SE	32	SN	21		

In our study, a matrix of $45 \times n$ represents chromosome with $T = m$, $C=n$, $P =8$ and $d=5$. For example teacher 5 will teach subject BI to class 1 at period 3,4 and subject PS to class 2 at period 5,6. In row 8 SN mean this subject taught as lesson in the classroom, the same subject SNL taught in lab , row 9.

3.2.3.2 Initialization

Based on the data that are provided we start our population randomly and at the same time checking necessary constraints for feasible timetable to exist. VB programming language is used to produce a matrix of $m \times n$ size, where m represent classes and n represent the number of time slot.

3.2.3.2.1 Initialization Procedure

- First, according to a standard of database provided by school management, we calculate the required number of teachers necessary to produce a timetable. Table (3.11,3.12)
- Calculate the free set of period p_i from total set P , $p_i = P - p_i$
- Calculate the unassigned teachers t_i and subjects s_i from total subjects S .
- Randomly assign pairs (s_i, t_i) to p_i and remove (p_i)
- All hard constraints must be considered to obtain feasible timetable.
- New time table generated must inherit the characteristics of the matrix $m \times n$.

Figure 3.3 shows population of chromosomes.

Table 311: Teacher Subject Relationship

Teacher	Subjects
3	BM,
6	BI, SE, AM, PM, AC
9	SE, AC, BC, CP
12	PM, AC, BC, CP, GE

Table 3.12: Class Subject Relationship.

Subject class	11	22	33	44	Total of lessons	No of teachers
BM	1	1	1	1	4	1
BI	2	3	1	1	7	1
SE	1	2	2	3	8	2
AM	1	1	2	3	7	1
PM	0	0	1	0	1	2
AC	0	0	1	0	1	3
BC	3	1	0	0	4	2
CP	0	0	0	0	0	2
GE	0	0	0	0	0	1

As we know the number of period is 8 a day for every class. Table 3.12 shows the total lessons of subject SE are 8 and we have only 2 teachers who can teach this subject see table 3.8. We distribute these lessons between teacher 6 and 9 which mean

that they can teach only another 4 subjects by each one. Now the next one is subject BI which is 7 lessons but we have only one teacher for this subject, teacher 6 who have already 4 periods, so this list of teachers is less than the minimum number of teachers required.

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	MA	27	BI	5	KHL	13	KHL	13	AM	6	AC	1	BM	25	BM	25
Class 2	BI	6	MA	27	BM	25	BM	25	SNL	7	SNL	7	SE	27	SE	27
Class 3	SNL	7	SNL	7	SE	6	BI	6	BC	12	PM	27	MA	2	GE	29
Class 4	KHL	13	KHL	13	MA	2	SN	7	AM	27	AC	12	BI	6	BI	6
Class 5	PN	10	MA	10	MA	10	EA	31	BM	32	BM	32	AM	11	AM	11
Class 6	PN	15	BI	11	PJ	8	PJ	8	PS	5	PS	5	PS	5	SN	21
Class 7	BI	8	BM	14	BM	14	PM	24	PS	19	PS	19	PS	19	MA	10
Class 8	MA	24	MA	24	SNL	28	SNL	28	PN	31	BI	11	AM	13	AM	13
Class 9	BI	11	EA	31	PJ	16	PJ	16	BM	14	BM	14	SNL	30	SNL	30
Class 10	BGL	30	BGL	30	BM	23	BM	23	MA	10	BI	22	AM	14	AM	14
Class 11	EA	31	PM	12	BI	11	SE	18	PD	17	PD	17	MA	24	MA	24
Class 12	PD	17	BI	8	PN	26	SN	26	MA	30	MA	30	BM	23	BM	23
Class 13	MA	28	GE	29	SE	9	SE	9	BI	8	BI	8	BM	3	BM	3
Class 14	PN	18	EA	15	PM	29	PM	29	BM	3	BM	3	BI	22	BI	22
Class 15	PS	19	PS	19	PS	19	EA	15	SNL	26	SNL	26	SE	9	SE	9
Class 16	BI	22	SN	26	PD	17	PD	17	MA	28	EA	15	EA	15	SE	18
Class 17	SNL	21	SNL	21	BI	22	SE	32	BM	18	BM	18	MA	28	MA	28

Figure 3.3: Population of School Timetable (two chromosomes)

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	KHL	13	AC	1	MA	27	BM	25	BM	25	BI	5	AM	6	KHL	13
Class 2	BM	25	SNL	7	BI	6	SE	27	SE	27	MA	27	SNL	7	BM	25
Class 3	SE	6	PM	27	SNL	7	GE	29	MA	2	SNL	7	BC	12	BI	6
Class 4	MA	2	AC	12	KHL	13	BI	6	BI	6	KHL	13	AM	27	SN	7
Class 5	MA	10	BM	32	PN	10	AM	11	AM	11	MA	10	BM	32	EA	31
Class 6	PJ	8	PS	5	PN	15	SN	21	PS	5	BI	11	PS	5	PJ	8
Class 7	BM	14	PS	19	BI	8	MA	10	PS	19	BM	14	PS	19	PM	24
Class 8	SNL	28	BI	11	MA	24	AM	13	AM	13	MA	24	PN	31	SNL	28
Class 9	PJ	16	BM	14	BI	11	SNL	30	SNL	30	EA	31	BM	14	PJ	16
Class 10	BM	23	BI	22	BGL	30	AM	14	AM	14	BGL	30	MA	10	BM	23
Class 11	BI	11	PD	17	EA	31	MA	24	MA	24	PM	12	PD	17	SE	18
Class 12	PN	26	MA	30	PD	17	BM	23	BM	23	BI	8	MA	30	SN	26
Class 13	SE	9	BI	8	MA	28	BM	3	BM	3	GE	29	BI	8	SE	9
Class 14	PM	29	BM	3	PN	18	BI	22	BI	22	EA	15	BM	3	PM	29
Class 15	PS	19	SNL	26	PS	19	SE	9	SE	9	PS	19	SNL	26	EA	15
Class 16	PD	17	EA	15	BI	22	SE	18	EA	15	SN	26	MA	28	PD	17
Class 17	BI	22	BM	18	SNL	21	MA	28	MA	28	SNL	21	BM	18	SE	32

Figure 3.3: Continued.

The following pseudo code describe how initial population created (figure 3.4)

1	While not all GP elements are placed
2	{
3	Randomly choose an element from GP
4	While the element is not Placed in the current chromosome
5	{
6	Randomly choose a Day
7	Randomly choose a Period
8	Try to Place the chosen GP element in the chromosome for 500 times
9	If not success
10	{
11	Try all empty genes
12	if not success Force placing it by swapping
13	}
14	}
15	}

Figure 3.4: Initial Population

In line 12, the proposed technique will try to insert the GP element regardless of the availability (whether it is occupied or empty) of the chosen place.

3.2.3.2.2 Evaluation

Once random population is created, each chromosome must be evaluated individually based on the set of soft constraints. The evaluation process is done by evaluation function, which calculates the fitness values of the chromosome. The fitness value is proportional to the sum of penalties of different constraint violation. Penalties are given systematically where for a constraint; higher penalty value is assigned for the most unwanted violations. To enhance the understanding of this topic, see example 1

In our project, we have 4 soft constraints listed bellow in a priority order.

- 1) Science subjects must be scheduled early in the timetable.

- 2) Lessons of the same subject for a class must be distributed uniformly over the week
- 3) Some lessons may only be scheduled to particular periods.
- 4) Teachers prefer to have more lessons on some days, in order to have a day without lessons (free day).

The total weight of these constraints is 100% and the weight of each constraint listed below depends on its property (table 3.13).

Table 3.13: Constraints and Values

No of constraint	Weight value
1	40
2	30
3	20
4	10

1) Constraint 1

For example the constraint 1 which is a science lesson should be scheduled early in the timetable. We calculate the total of all science lessons over the entire chromosome, then we divided the weight value on the number science lessons. then we come out with individual penalty value for each one by giving a rate for each period for example

If we have 32 science lessons per week, by default we know that the weight of science

lessons is 40%. So the fitness value for each one is $\frac{40}{32} = 1.25$ meaning that if the math

lesson comes at the first period we give it this value (1.25) but if it comes at period three its value will be (0.892857144). The constraint value is reduced according to its position

in the timetable $\frac{1.25}{7} = 0.178571428$ (see table 3.14). For example:

Table 3.14: Value Distributed Over Periods.

Period	Lesson1 value
1	1.25
2	1.071428572
3	0.892857144
4	0.714285716
5	0.535714288
6	0.35714286
7	0.178571432
8	0.000000000

To calculate the fitness of this constraint overall the timetable we use the following formula.

n is 1, 2, 3, 4, V_c is the value of a constraint and w_c is weight it.

$$\text{Constraint fitness } (c_n) = \left(\sum_{i=1}^k V_c \right) \times w_c \quad (1)$$

This is only to come out with the fitness of one constraint over the timetable. Formula (2) calculates fitness of the entire constraints over the timetable as follows:

n is number of constraints, c_n is constraint fitness, and t_c is chromosome fitness

$$\text{Chromosome fitness } (t_c) = \sum_{i=1}^n c_n \quad (2)$$

2) Constraint 2

Lessons of the same subject for a class should be distributed uniformly over the week. The weight value of this constraint depends on its priority is 30. if we have 5 classes each class have 10 subject to be taught weekly we assign value to each class $\frac{30}{5} = 6$. 6 is the value of entire subject taught for class. Value of each subject is

$\frac{6}{10} = 0.6$. 0.6 is value of one subject for one class if the subject distributed overall the

week on other words if class x study subject MA 5 period a week each day have lesson, value of this subject will be 0.6. see table

Table 3.15: Value Distributed Over Days.

Day	Lesson value
1	0
2	0.2
3	0.3
4	0.5
5	0.6

Total value of specific subject for class calculated as shown if formula (3)

SV is subject value, S_n is subject

$$SV = \sum_{day1}^5 S_n \quad (3)$$

Total value of all subject taught to a class calculated as follow

CSV represent entire subjects value taught to a class

$$CSV = \sum_{day1}^5 SV \quad (4)$$

WT represent the total values of this constraint W is the weight of constraints overall the timetable.

$$WT = \sum_{day1}^5 CSV \times W \quad (5)$$

3) Constraint 3

Calculation of constraint 4 as same as constraint 1

4) Constraint 4

Teachers prefer to have more lessons on some days, in order to have a day without lessons (free day). calculation of this constraint is very simple because teacher may have one day free if possible. The weight of this constraint is 20. if we have 20 teacher in the school then teacher who have one free day will valuated by 1 and teacher who have no free day valuated by 0. See formula (6)

DF represent value of day free for a teacher, c class, tv value of teacher, and W the weight of constraint

$$DF = \sum_{c_1}^{c_n} c \times \sum_{t_1}^{t_n} tv \times W \quad (6)$$

3.2.3.3 Genetic Algorithms Operators

3.2.3.3.1 Chromosomes Selection

There are many kinds of parent selection techniques, such as fitness-based selection (roulette wheel parent selection), rank-based selection, tournament-based selection, and spatially oriented selection. (Michalewicz, 1999) In this project, we use two independently selection techniques which are:

A: Roulette Wheel Parent Selection.

Each chromosome has a chance of being selected that is directly proportional to its fitness. The effect of this depends strongly on the range of fitness values in the current population. This technique is complicated but it gives more reproductive chances, in total, to those population members that the most fit.

B: Rank Based Selection.

This method was reported by (Baker, 1985) where selection probabilities were based on a chromosome's relative rank or position in the population, rather than its fitness. Rank selection ranks the population first and then every chromosome receives a fitness value determined by this ranking. The worst ranked chromosome will receive a fitness value 1 , the second worst is 2 and so on. The best will have a fitness N (this is the number of chromosomes in the population).

3.2.3.2.2 Crossover

Two chromosomes are cut (dissected) at a selected point at random to make head and tail portions. The dissected parts are then switched around where the head of the first chromosome is combined with the tail of the second one and vice versa creating two new chromosomes. Figure 3.5 shows the crossover on the timetable.

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	MA	27	BI	5	KHL	13	KHL	13	AM	6	AC	1	BM	25	BM	25
Class 2	BI	6	MA	27	BM	25	BM	25	SNL	7	SNL	7	SE	27	SE	27
Class 3	SNL	7	SNL	7	SE	6	BI	6	BC	12	PM	27	MA	2	GE	29
Class 4	KHL	13	KHL	13	MA	2	SN	7	AM	27	AC	12	BI	6	BI	6
Class 5	PN	10	MA	10	MA	10	EA	31	BM	32	BM	32	AM	11	AM	11
Class 6	PN	15	BI	11	PJ	8	PJ	8	PS	5	PS	5	PS	5	SN	21
Class 7	BI	8	BM	14	BM	14	PM	24	PS	19	PS	19	PS	19	MA	10
Class 8	MA	24	MA	24	SNL	28	SNL	28	PN	31	BI	11	AM	13	AM	13
Class 9	BI	11	EA	31	PJ	16	PJ	16	BM	14	BM	14	SNL	30	SNL	30
Class 10	BGL	30	BGL	30	BM	23	BM	23	MA	10	BI	22	AM	14	AM	14
Class 11	EA	31	PM	12	BI	11	SE	18	PD	17	PD	17	MA	24	MA	24
Class 12	PD	17	BI	8	PN	26	SN	26	MA	30	MA	30	BM	23	BM	23
Class 13	MA	28	GE	29	SE	9	SE	9	BI	8	BI	8	BM	3	BM	3
Class 14	PN	18	EA	15	PM	29	PM	29	BM	3	BM	3	BI	22	BI	22
Class 15	PS	19	PS	19	PS	19	EA	15	SNL	26	SNL	26	SE	9	SE	9
Class 16	BI	22	SN	26	PD	17	PD	17	MA	28	EA	15	EA	15	SE	18
Class 17	SNL	21	SNL	21	BI	22	SE	32	BM	18	BM	18	MA	28	MA	28

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	KHL	13	AC	1	MA	27	BM	25	BM	25	BI	5	AM	6	KHL	13
Class 2	BM	25	SNL	7	BI	6	SE	27	SE	27	MA	27	SNL	7	BM	25
Class 3	SE	6	PM	27	SNL	7	GE	29	MA	2	SNL	7	BC	12	BI	6
Class 4	MA	2	AC	12	KHL	13	BI	6	BI	6	KHL	13	AM	27	SN	7
Class 5	MA	10	BM	32	PN	10	AM	11	AM	11	MA	10	BM	32	EA	31
Class 6	PJ	8	PS	5	PN	15	SN	21	PS	5	BI	11	PS	5	PJ	8
Class 7	BM	14	PS	19	BI	8	MA	10	PS	19	BM	14	PS	19	PM	24
Class 8	SNL	28	BI	11	MA	24	AM	13	AM	13	MA	24	PN	31	SNL	28
Class 9	PJ	16	BM	14	BI	11	SNL	30	SNL	30	EA	31	BM	14	PJ	16
Class 10	BM	23	BI	22	BGL	30	AM	14	AM	14	BGL	30	MA	10	BM	23
Class 11	BI	11	PD	17	EA	31	MA	24	MA	24	PM	12	PD	17	SE	18
Class 12	PN	26	MA	30	PD	17	BM	23	BM	23	BI	8	MA	30	SN	26
Class 13	SE	9	BI	8	MA	28	BM	3	BM	3	GE	29	BI	8	SE	9
Class 14	PM	29	BM	3	PN	18	BI	22	BI	22	EA	15	BM	3	PM	29
Class 15	PS	19	SNL	26	PS	19	SE	9	SE	9	PS	19	SNL	26	EA	15
Class 16	PD	17	EA	15	BI	22	SE	18	EA	15	SN	26	MA	28	PD	17
Class 17	BI	22	BM	18	SNL	21	MA	28	MA	28	SNL	21	BM	18	SE	32

Figure 3.5: Offspring Timetable

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	MA	27	BI	5	KHL	13	KHL	13	AM	6	AC	1	BM	25	BM	25
Class 2	BI	6	MA	27	BM	25	BM	25	SNL	7	SNL	7	SE	27	SE	27
Class 3	SNL	7	SNL	7	SE	6	BI	6	BC	12	PM	27	MA	2	GE	29
Class 4	KHL	13	KHL	13	MA	2	SN	7	AM	27	AC	12	BI	6	BI	6
Class 5	PN	10	MA	10	MA	10	EA	31	BM	32	BM	32	AM	11	AM	11
Class 6	PN	15	BI	11	PJ	8	PJ	8	PS	5	PS	5	PS	5	SN	21
Class 7	BI	8	BM	14	BM	14	PM	24	PS	19	PS	19	PS	19	MA	10
Class 8	SNL	28	BI	11	MA	24	AM	13	AM	13	MA	24	PN	31	SNL	28
Class 9	PJ	16	BM	14	BI	11	SNL	30	SNL	30	EA	31	BM	14	PJ	16
Class 10	BM	23	BI	22	BGL	30	AM	14	AM	14	BGL	30	MA	10	BM	23
Class 11	BI	11	PD	17	EA	31	MA	24	MA	24	PM	12	PD	17	SE	18
Class 12	PN	26	MA	30	PD	17	BM	23	BM	23	BI	8	MA	30	SN	26
Class 13	SE	9	BI	8	MA	28	BM	3	BM	3	GE	29	BI	8	SE	9
Class 14	PM	29	BM	3	PN	18	BI	22	BI	22	EA	15	BM	3	PM	29
Class 15	PS	19	SNL	26	PS	19	SE	9	SE	9	PS	19	SNL	26	EA	15
Class 16	PD	17	EA	15	BI	22	SE	18	EA	15	SN	26	MA	28	PD	17
Class 17	BI	22	BM	18	SNL	21	MA	28	MA	28	SNL	21	BM	18	SE	32

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	KHL	13	AC	1	MA	27	BM	25	BM	25	BI	5	AM	6	KHL	13
Class 2	BM	25	SNL	7	BI	6	SE	27	SE	27	MA	27	SNL	7	BM	25
Class 3	SE	6	PM	27	SNL	7	GE	29	MA	2	SNL	7	BC	12	BI	6
Class 4	MA	2	AC	12	KHL	13	BI	6	BI	6	KHL	13	AM	27	SN	7
Class 5	MA	10	BM	32	PN	10	AM	11	AM	11	MA	10	BM	32	EA	31
Class 6	PJ	8	PS	5	PN	15	SN	21	PS	5	BI	11	PS	5	PJ	8
Class 7	BM	14	PS	19	BI	8	MA	10	PS	19	BM	14	PS	19	PM	24
Class 8	MA	24	MA	24	SNL	28	SNL	28	PN	31	BI	11	AM	13	AM	13
Class 9	BI	11	EA	31	PJ	16	PJ	16	BM	14	BM	14	SNL	30	SNL	30
Class 10	BGL	30	BGL	30	BM	23	BM	23	MA	10	BI	22	AM	14	AM	14
Class 11	EA	31	PM	12	BI	11	SE	18	PD	17	PD	17	MA	24	MA	24
Class 12	PD	17	BI	8	PN	26	SN	26	MA	30	MA	30	BM	23	BM	23
Class 13	MA	28	GE	29	SE	9	SE	9	BI	8	BI	8	BM	3	BM	3
Class 14	PN	18	EA	15	PM	29	PM	29	BM	3	BM	3	BI	22	BI	22
Class 15	PS	19	PS	19	PS	19	EA	15	SNL	26	SNL	26	SE	9	SE	9
Class 16	BI	22	SN	26	PD	17	PD	17	MA	28	EA	15	EA	15	SE	18
Class 17	SNL	21	SNL	21	BI	22	SE	32	BM	18	BM	18	MA	28	MA	28

Figure 3.5: Continued.

3.2.3.2.3 Mutation

The mutation operator should be random, and yet it must not assign any time slot not allowed for that particular class. This is solved by first reading in all the allowable time slots for each class and mapping the random gene to the length of that array and then assigning the content to the selected random gene. It would be as follows:

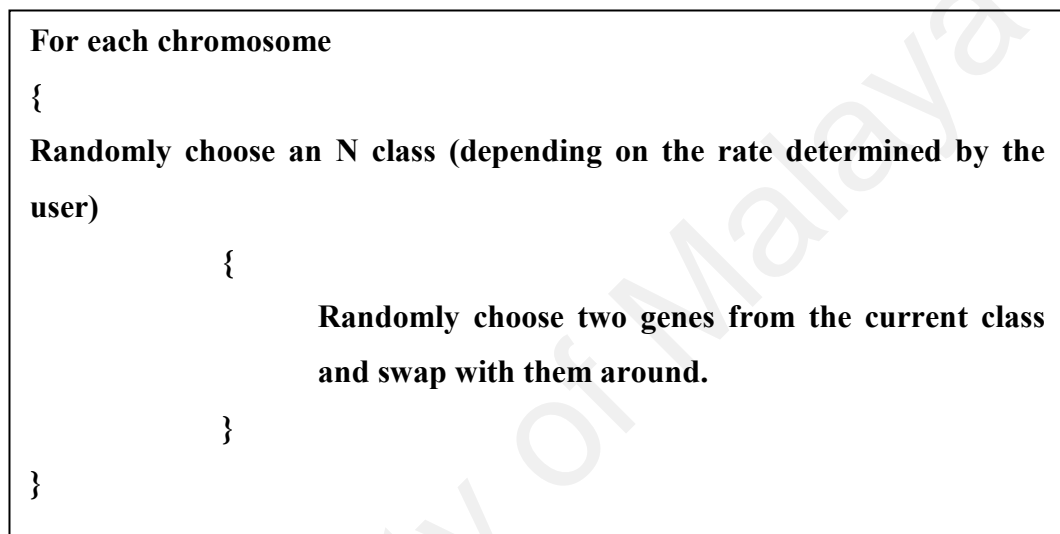


Figure 3.6: Mutation Process

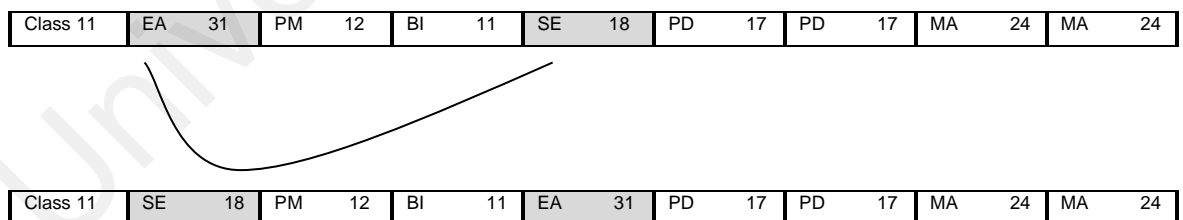


Figure 3.7: Two Genes Mutation

Mutation is performed on randomly selected classes where two or more periods (time slot) replaced with each other. Mutation, like crossover, must also ensure that a timetable remains feasible (hard constraints are satisfied) after its action. It cannot,

therefore, take any lesson and shift it to another period at random, since this may cause a conflict between the moved lesson and ones which have been already scheduled.

University of Malaya

Chapter 4: GA Design and Implementation

4.1 Introduction

The user in our problem will be a representative sample of one of the Malaysian secondary schools. The user will have to provide information about teachers, subjects, lab room and year of study. And as mentioned in the first chapter, students and classrooms will be prefixed all the time. But firstly, it is necessary to define the system in the context of data flow diagrams. (Figure 4.1) shows only a one to one relationship between the user and the system.

4.2 Context Diagrams

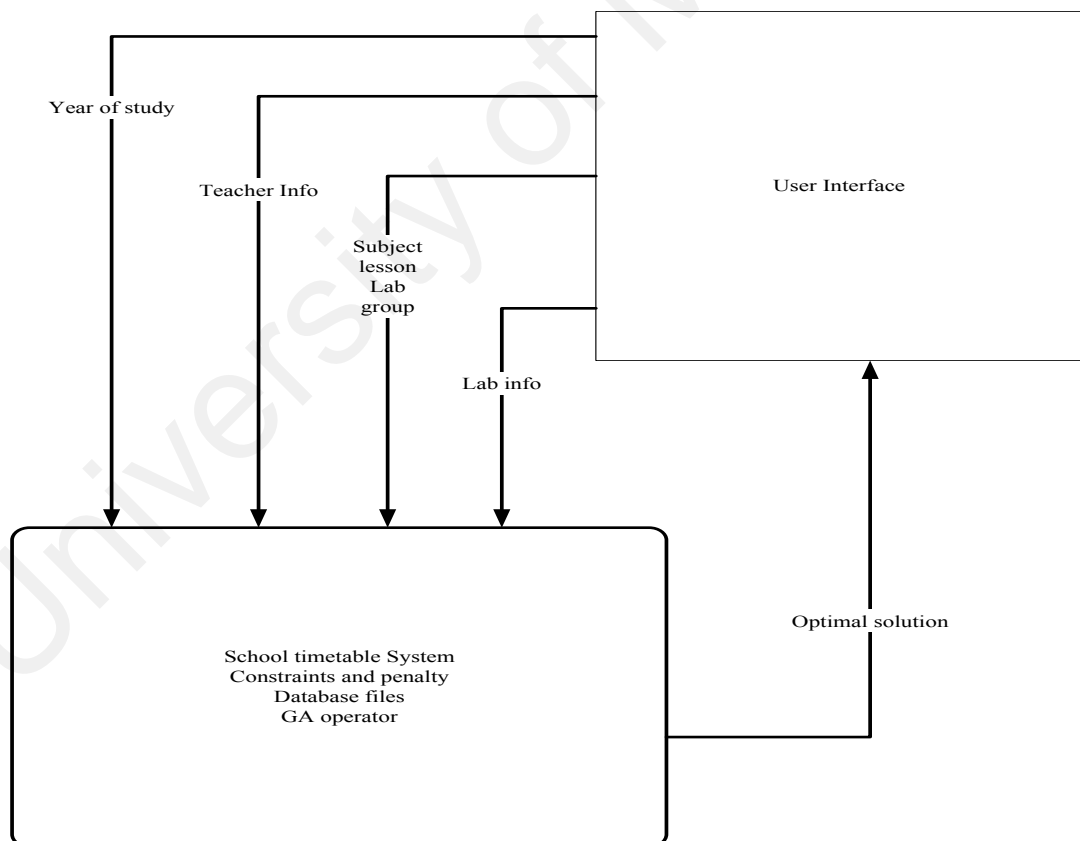


Figure 4.1: Context Diagram

4.2.1 Year of Study

Malaysia secondary schools are 5 years study system. The first and the second years have no elective subjects but the other three years have different elective subjects depending on the study year.

4.2.2 Teacher Information

It contains names of teachers and the subjects they are handling. This is necessary for predicting the teacher clashes.

4.2.3 Subject Information

It contains detailed information about a subject such as the number of lessons for each class, number of labs, and whether it is group subject or not. This is crucial for chromosome encoding. Each subject attached with three options: lesson, lesson and lab, and group only.

- **Lesson**

Most of secondary school subjects are taught at the classroom this must be clarified by the user to ensure that it is not attached to a laboratory. Table 4.1 shows some sample subjects which can only be handled in classroom.

Table 4.1: Lessons Handled at Classroom.

No	Short name	Subject
1	BM	Bahasa Melayu
2	BI	Bahasa Inggeris
3	EA	Ekonomi Asas
4	PN	Pengkayaan
5	PM	Pendidikan Moral

- **Lab**

A subject of this type is usually taught as a lesson at the classroom followed by a practical training in the lab and it is not necessary to be handled at the same day, although lab lessons preferred to be two continual periods. Table 4.2 shows some sample lab subjects

Table 4.2: Subject Handled at Lab.

No of rooms	Short name	Subject
3	SN	Sains
2	PJ	Pend Jasmani & Kesihatan
1	FZ	Fizik
1	BG	Biologi
5	KM	Kimia

- **Group**

This is the feature that makes our system unique. Group subject can be:

1. Students from two different classrooms in same study year are combined together and then divide into two groups to be taught different subjects according to some conditions (i.e., religious study).
2. Students from the same classroom divided into two groups to be taught different subjects. This is usually a sex-based segregation (boys and girls).

4.3 Data Flow Diagram

Figure 4.2 expands the context diagram and shows two main building blocks of the school timetable system, the user interface and the genetic algorithm. These two are absolutely separated sub-systems where block 1 caters for the need of data collection from the user and data display to the user. Block 2 is invisible and functions as the core of the processing of the system. The data storages are the only interface between block 1 and block 2.

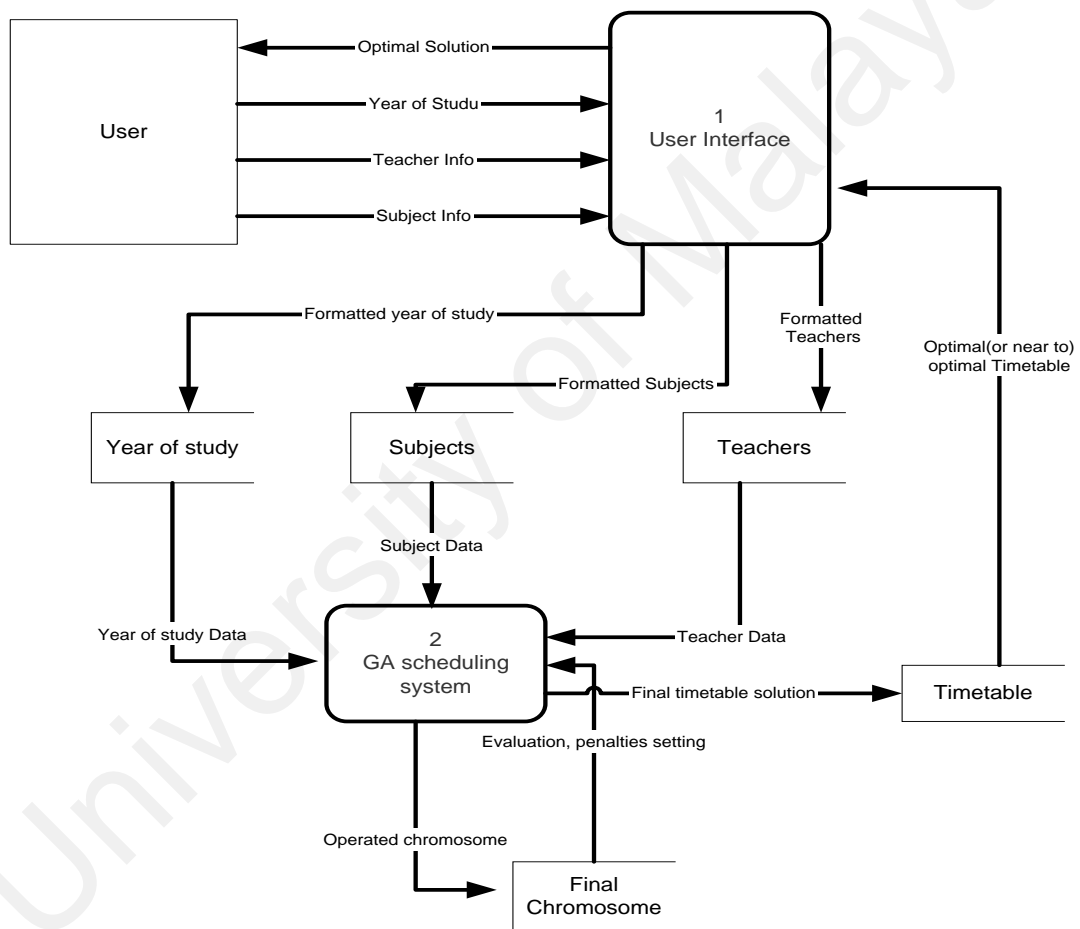


Figure 4.2: General View of Data Flow

Since block 1 in figure 4.2 is very simple and has no possible expansions, we directly move to data flow diagram (DFD) for block 2. Figure 4.3 displays the data

flow between the sub processes in the Malaysia school timetable using GA (MSTT-GA). These sub-processes will be discussed in great details through out this chapter.

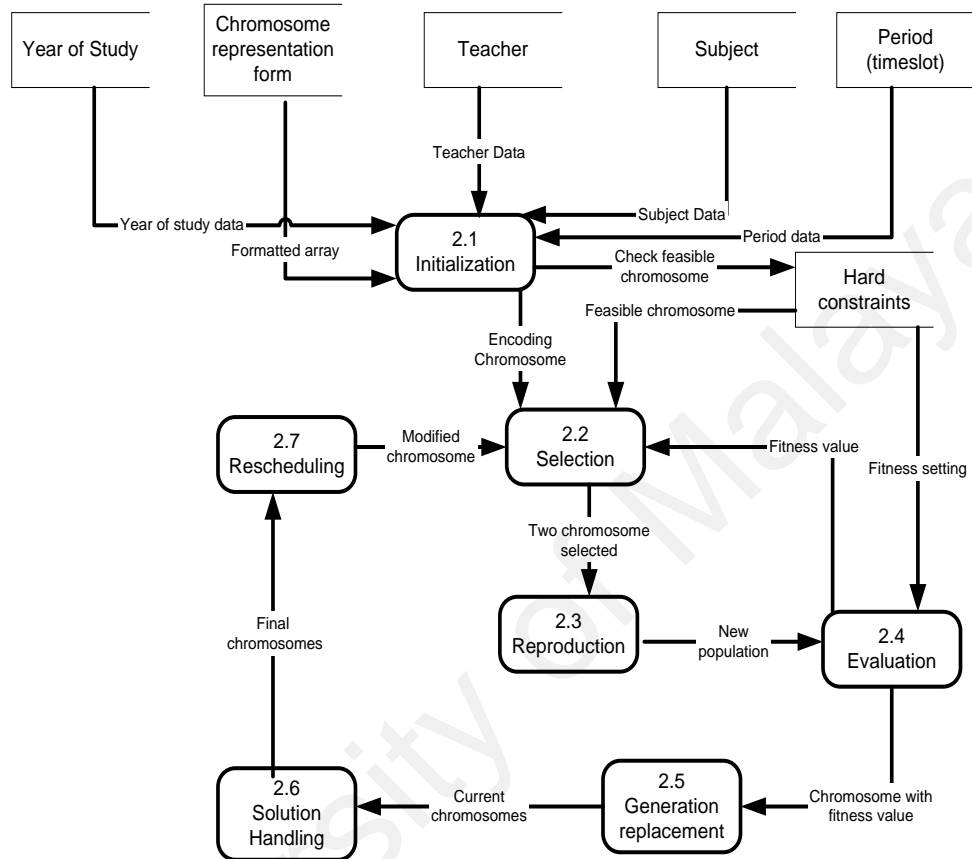


Figure 4.3: View Detailed of Block 2

4.4 Initialization

Input files of the system are given as a text file (table 4.3 and 4.4). For example, subject MA taught to class 1 5 period per week (table 4.3). Subject MA taught by teachers 2, 10, 24, 28, and 30 (table 4.4). This part represents the database files or the first part in figure 4.3:

Table 4.3: Relationship between Subjects and Classes.

"MA"	"KH"	"EA"	"BM"
1, 5	4		6
2, 5	4		6
3, 5	4		6
4, 5	4		6
5, 5		3	6
6, 5		3	6
7, 5		3	6
8, 5		3	6
9, 5		3	6
10, 5			6
11, 5		3	6
12, 5		3	6
13, 5			6
14, 5		3	6
15, 5		3	6
16, 5		3	6
17, 5		3	6
"BI"	"PN"	"PD"	"PM"
1, 5			
2, 5			3
3, 5			3
4, 5			
5, 5	2	3	
6, 5	3	3	3
7, 5			3

Table 4.4: Relationship between Subjects and Teachers.

1, AC	17, PD, PN
2, MA	18, PN, SE, BM
3, BM	19, PS, AM, PN
4, AC, AM	20, KH
5, BI, PS	21, SN, FZ, PN
6, BI, SE, AM, PM, AC	22, BI, AC
7, SN, PJ,	23, BM, PN
8, BI, PJ, CP	24, MA, PM
9, SE, AC, BC, CP	25, BM, CP
10, MA, PN, MT	26, SN, PN, KM
11, BI, AM, CP, PS	27, MA, AM, PM, SE, PN
12, PM, AC, BC, CP, GE	28, MA, SN, AC
13, KH, AM, BM	29, GE, PM, PJ, AM, PN
14, BM, AM, AC, PM	30, BG, MA, SN, PN
15, PN, EA	31, EA, PN, PA
16, PX, BC, PN	32, SE, BM

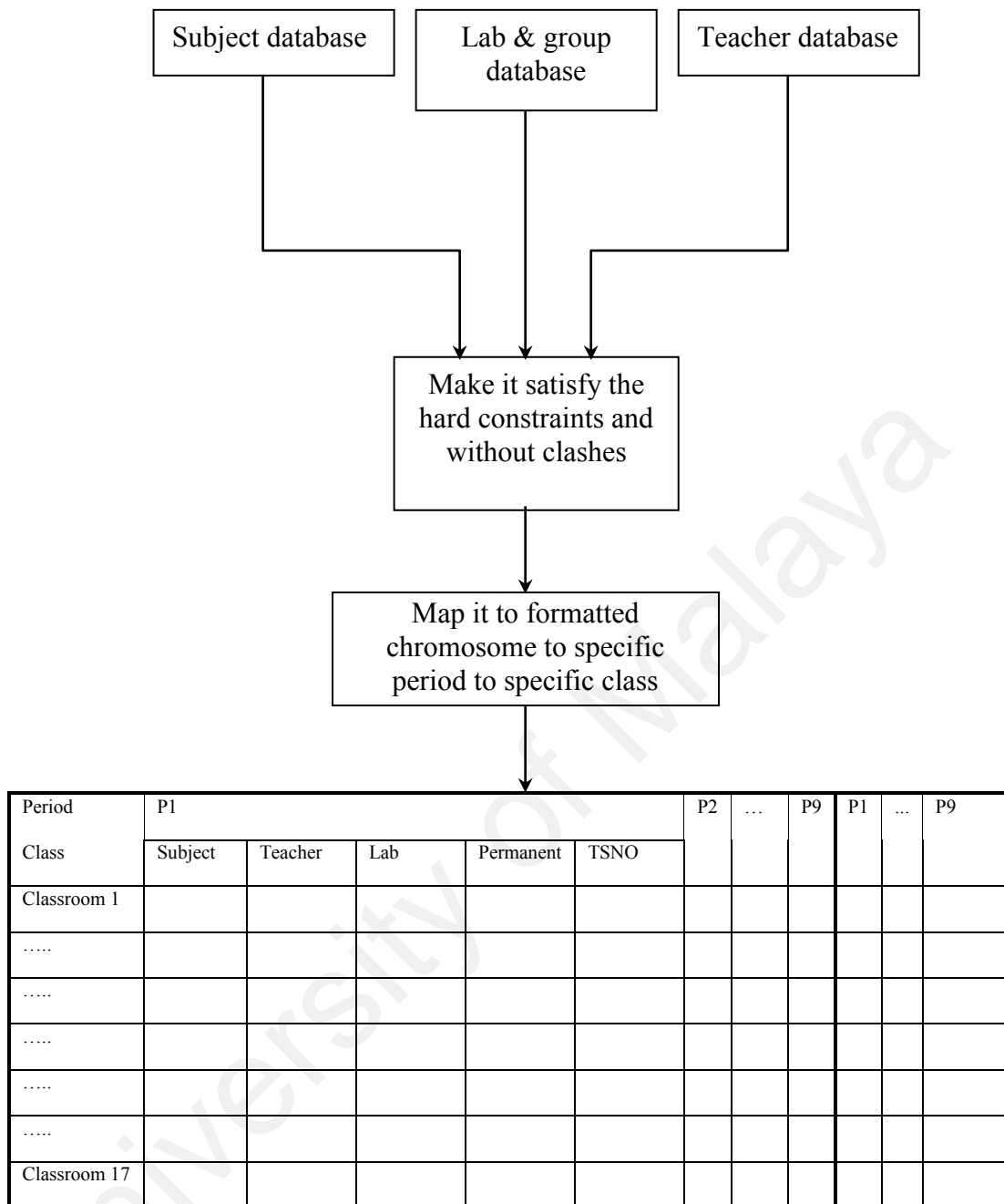


Figure 4.4: Mapping Data to Chromosome

Figure 4.4 represent the process of initialization figure 4.3. It assigns teacher and subject to a particular class in a particular period of time and day (time slot) without any clash. Table 4.5 shows the final result of initialization. Permanent and TSNO are special fields used when the user selects population to be generated randomly. TSNO represents the position of an element in the Teacher Subject array, when putting the elements

randomly into the chromosome. The “Generate population” routine attempts up to 500 times to get an acceptable position, but when it fails to find an acceptable position the entire chromosome will be canceled and generate new one.

Table 4.5: Result of Initialization

	P1		P2		P3		P4		P5		P6		P7		P8	
Class 1	MA	27	BI	5	KHL	13	KHL	13	AM	6	AC	1	BM	25	BM	25
Class 2	BI	6	MA	27	BM	25	BM	25	SNL	7	SNL	7	SE	27	SE	27
Class 3	SNL	7	SNL	7	SE	6	BI	6	BC	12	PM	27	MA	2	GE	29
Class 4	KHL	13	KHL	13	MA	2	SN	7	AM	27	AC	12	BI	6	BI	6
Class 5	PN	10	MA	10	MA	10	EA	31	BM	32	BM	32	AM	11	AM	11
Class 6	PN	15	BI	11	PJ	8	PJ	8	PS	5	PS	5	PS	5	SN	21
Class 7	BI	8	BM	14	BM	14	PM	24	PS	19	PS	19	PS	19	MA	10
Class 8	MA	24	MA	24	SNL	28	SNL	28	PN	31	BI	11	AM	13	AM	13
Class 9	BI	11	EA	31	PJ	16	PJ	16	BM	14	BM	14	SNL	30	SNL	30
Class 10	BGL	30	BGL	30	BM	23	BM	23	MA	10	BI	22	AM	14	AM	14
Class 11	EA	31	PM	12	BI	11	SE	18	PD	17	PD	17	MA	24	MA	24
Class 12	PD	17	BI	8	PN	26	SN	26	MA	30	MA	30	BM	23	BM	23
Class 13	MA	28	GE	29	SE	9	SE	9	BI	8	BI	8	BM	3	BM	3
Class 14	PN	18	EA	15	PM	29	PM	29	BM	3	BM	3	BI	22	BI	22
Class 15	PS	19	PS	19	PS	19	EA	15	SNL	26	SNL	26	SE	9	SE	9
Class 16	BI	22	SN	26	PD	17	PD	17	MA	28	EA	15	EA	15	SE	18
Class 17	SNL	21	SNL	21	BI	22	SE	32	BM	18	BM	18	MA	28	MA	28

Table 4.5 is a real timetable representing one day with 8 periods for 17 classes. The number of generations is decided by the user. The following is the initialization algorithm (Figure 4.5)

```

For each chromosome in the population do
  For each class do
    For each period do
      Assign a subject with its teacher that does not cause any violation to
      the hard constraints to the period
  
```

Figure 4.5 Initialize Population Algorithms.

4.4.1 Calculating the Clashes:-

The main objective of this procedure is to calculate the clashes among the teachers as follows:

1. Read each gene (teacher and subject) on each row on the timetable.
2. Compare read gene with others on the same column. Place that gene if there is no identical one.
3. If so, try with next gene.
4. Repeat it until the timetable is full and usable.

4.5 Genetic Algorithm Modules

Figure 4.3 shows the GA components (2.2, 2.3, and 2.4) which are Selection, Reproduction and Evolution to represent the simple GA implemented in this project. These operators have been described in great detail in the previous chapter. In this project we chose the most popular selection techniques Roulette Wheel and Rank selection, to be implemented with this system for obvious reasons. As illustrated in figure 4.3, the encoding chromosome is received by one of the selection techniques, Roulette Wheel or Rank based selection depending on the user. Once selection is completed then the two selected chromosomes are sent to the second GA technique, reproduction (2.3). This component includes two GA operators which are crossover and mutation. User has the option of crossover type (one or two points). The last step is evolution (2.4) and assigning fitness to each chromosome of the new population. At this point, user can stop implementing the system and consider the current best chromosome as final solution. The following pseudo code shows how GA operators work in this system.

4.6 Selection Module

- **Roulette wheel**

The Roulette wheel parent selection is one of the parent selection techniques that select the parent randomly from the population. The following pseudo code is the Roulette wheel parent selection algorithm (see figure 4.6).

```
Roulette wheel parent selection
{
    • Calculate the sum of all the fitness of population members. It is equal to the cumulative fitness of the last population member. The cumulative fitness is the fitness of chromosome added to the fitness of preceding population members.
    • Generate r, a random number between zero and the total fitness.
    • Return the first population member whose current cumulative total fitness is greater than or equal to r.
}
```

Figure 4.6: Roulette Wheel Parent Selection

- **Rank based**

The chromosomes in the population are sorted according to fitness value then allocated the number of offspring they will produce based on their ranking in the population. The main advantage of this method is that it uses relative fitness information, ignoring absolute values and thus removes any need to scale fitness scores.

4.7 Crossover Module

- **One Point Crossover**

As mentioned earlier in chapter three, two chromosomes are chosen randomly. In one point crossover operator implementation, swap the first part of the first chromosome with the last part of the second chromosome and vice versa. The following pseudo code represents one point crossover algorithm (see figure 4.7).

```
{  
  • Choose two parents from population using roulette wheel or rank based  
  • Determine one random point to be chosen for swapping  
  • For the first point until N  
    ▪ Choose random gene = random number from first gene until last gene.  
    ▪ Swap the random gene between parents.  
  • Check the time slot to make sure it is free of clashes and fulfill other constraints after swapping.  
  • Calculate the fitness of child.  
  • Find the worst member in population  
  • If fitness of child is greater than fitness of worst member  
    ▪ Insert the child into next generation.  
}
```

Figure 4.7: One Point Crossover Algorithm

- **Two Point Crossover**

The procedures of two point crossover are as the same as one point. The major difference between them is that the two point crossover operates with two random points as described in the following algorithms (see figure 4.8)

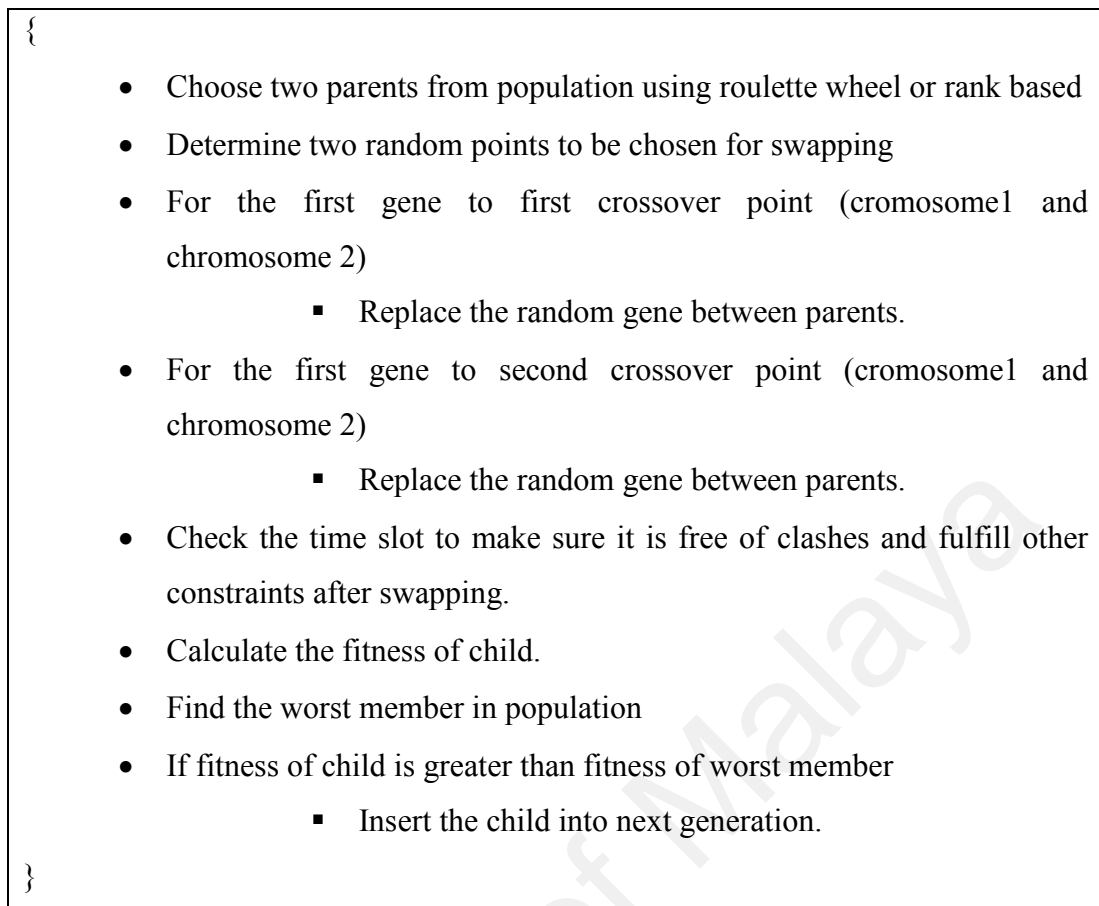


Figure 4.8: Two Point Crossover Algorithms

4.8 Mutation Module

In the mutation operator, the gene representing teacher and subject is replaced randomly by another gene if the predefined probability rate is passed in the given chromosome.

The following pseudo code is the mutation algorithm (see figure 4.9).

Mutation

{

- Choose one parent by using roulette wheel or rank based selection.
- For first gene to last gene
 - Generate one random number, r between 0 and 1 for every gene.
 - If r less than mutation probability parameter
 - Generate new gene randomly and replace the current gene
 - Check the timeslot is free of clashes and fulfill overflow constraint after mutation
 - Calculate the fitness of mutation child
 - Find the worst member (lowest fitness) among the population
 - If fitness of mutation child is greater than fitness of worst member
 - Insert the mutation child into next generation.

}

Figure 4.9: Mutation Algorithm

4.9 The Fitness Function and Penalties

The fitness function is a vital component of any genetic algorithm implementation. It is the values returned by the fitness function upon evaluating an individual that guide the otherwise stochastic algorithm through the search space.

The fitness score of a chromosome reveals how good the solution it represents is. The penalty of a chromosome is the opposite: it is a measure of negative utility, of the degree to which it violates constraints. Penalty and fitness thus have an inverse relationship:

$$Fitness(t_c) = \sum_{i=1}^n c_n$$

If $P(i)$ represents the penalty score of the timetable defined by chromosome i and pc the penalty associated with a violation of constraint C , the constraints described in the previous section can be thought of (and are easily implemented as) a series of conditional statements of the form:

If constraint C is violated by

$$i P(i) := P(i) + pc$$

End If

Penalty values are described in previous chapter.

4.10 User interface

4.10.1 User data entry

This part concentrates on the primary original data provided by school management as database for MSTTP-GA. Figure 10 is the main user data entry form. The form consists of six layers: Subjects; Classes; Teachers; Class/Subject; Teacher/Subject; and Labs.

Layer 1: Subjects

This layer help user to enter a subject name and some details about it (group subject, lab lesson, or science). The layer provided by List box to allow the user to check the entered data, delete or modify it (see figure 4.10).

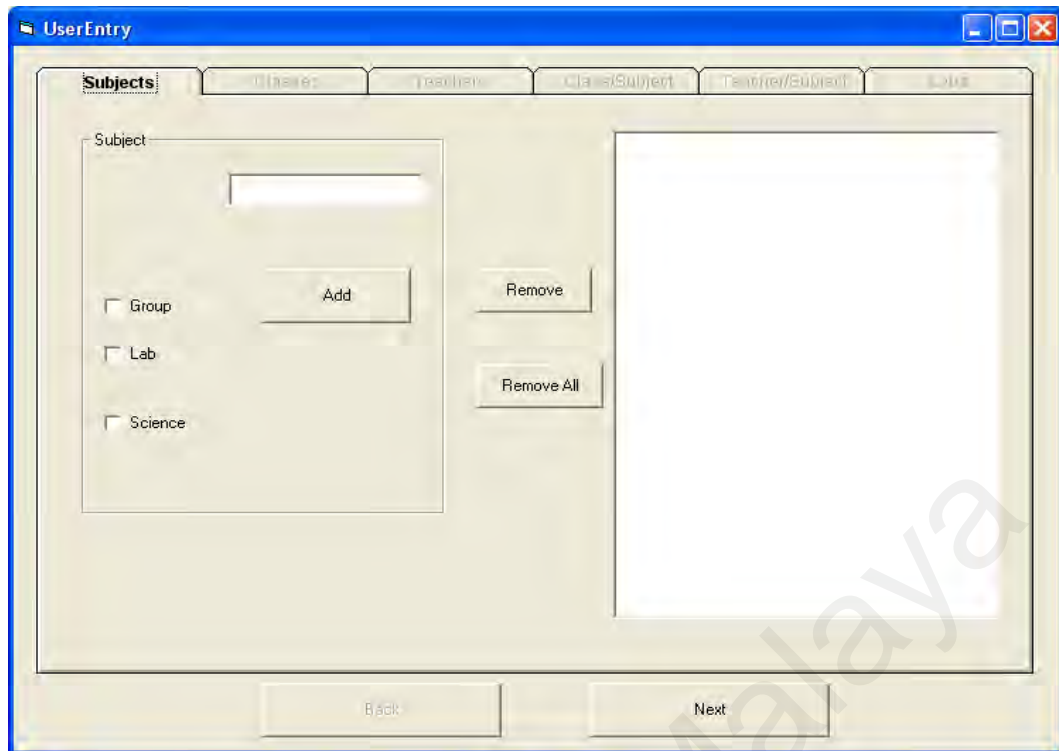


Figure 4.10: Subjects Layer.

Layer 2: Classes

The two main objects of this layer are the two text box, the first one represent year of study and the second one represent classroom where the students learning their lessons (see figure 4.11).

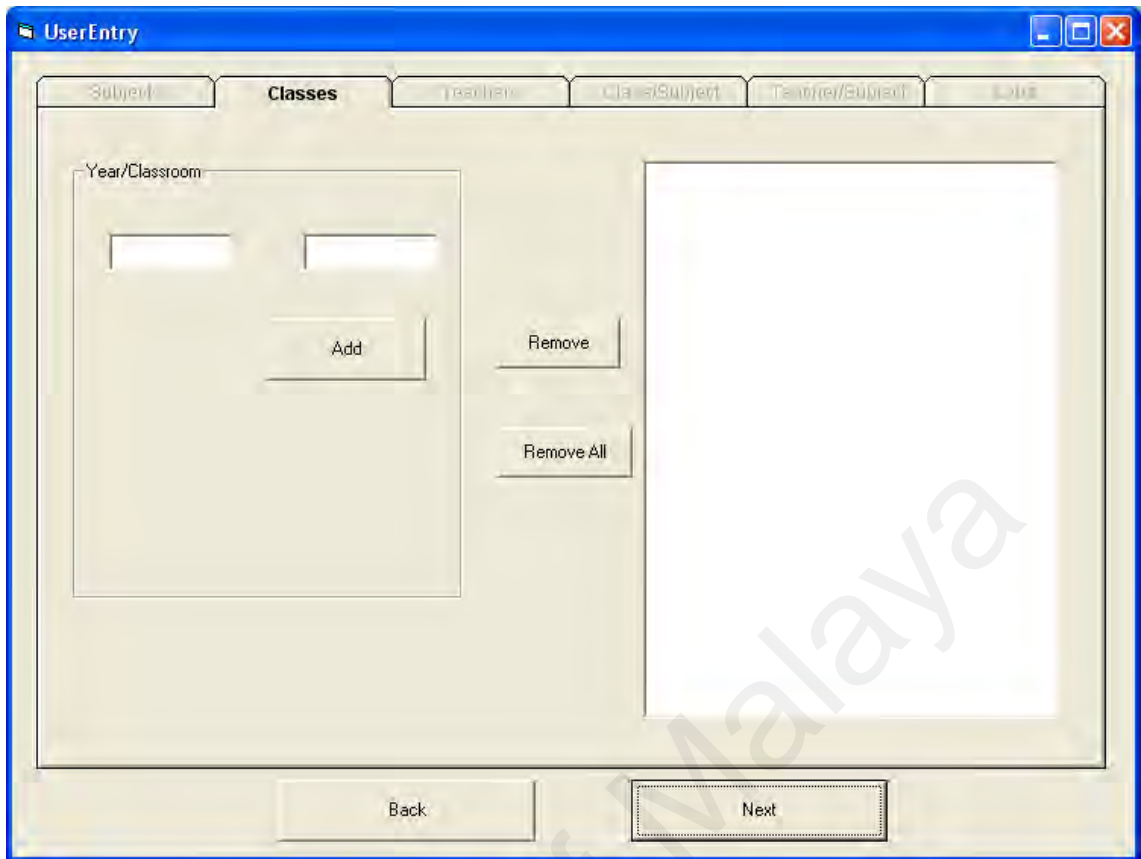


Figure 4.11: Classes Layer.

Layer 3: Teachers

Teacher layer help user to enter teachers name to database of system (see figure 4.12).

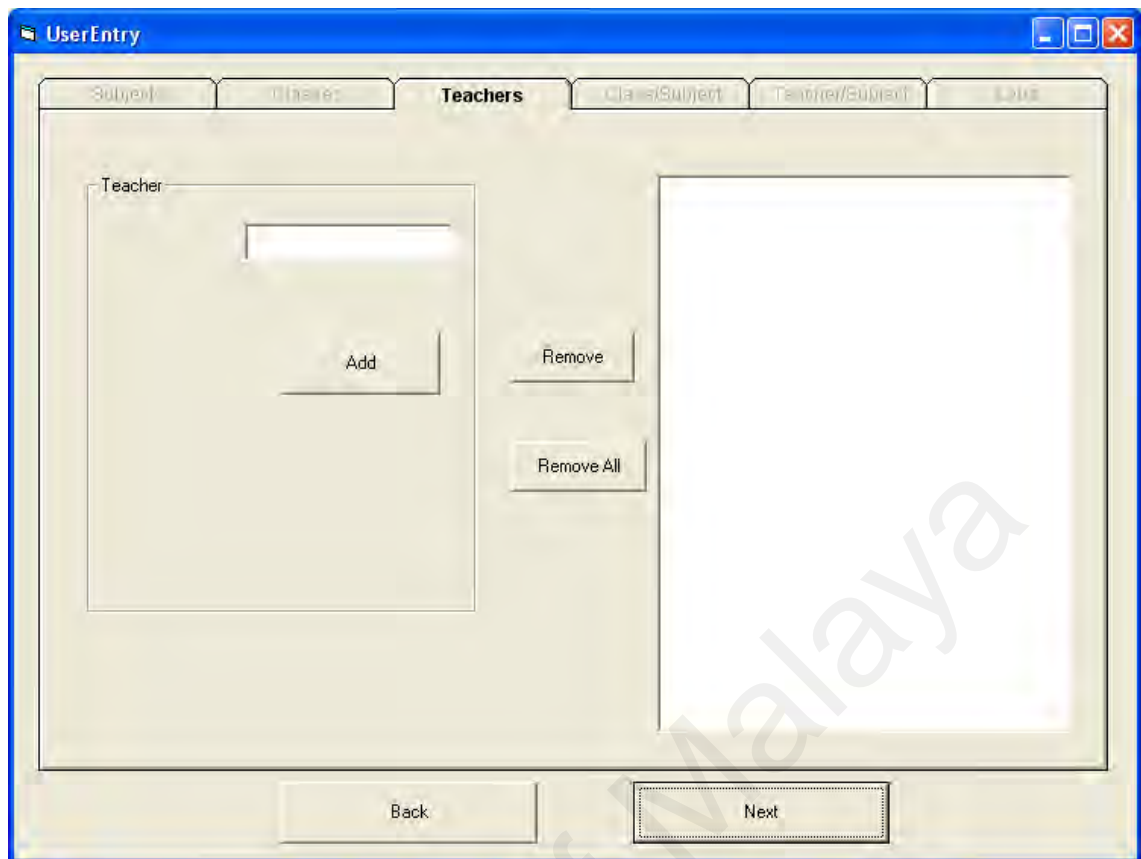


Figure 4.12: Teachers Layer.

Layer 4: Class/Subject

This layer is different from the other in that list of previous entered classes and subject match together (see figure 4.13).

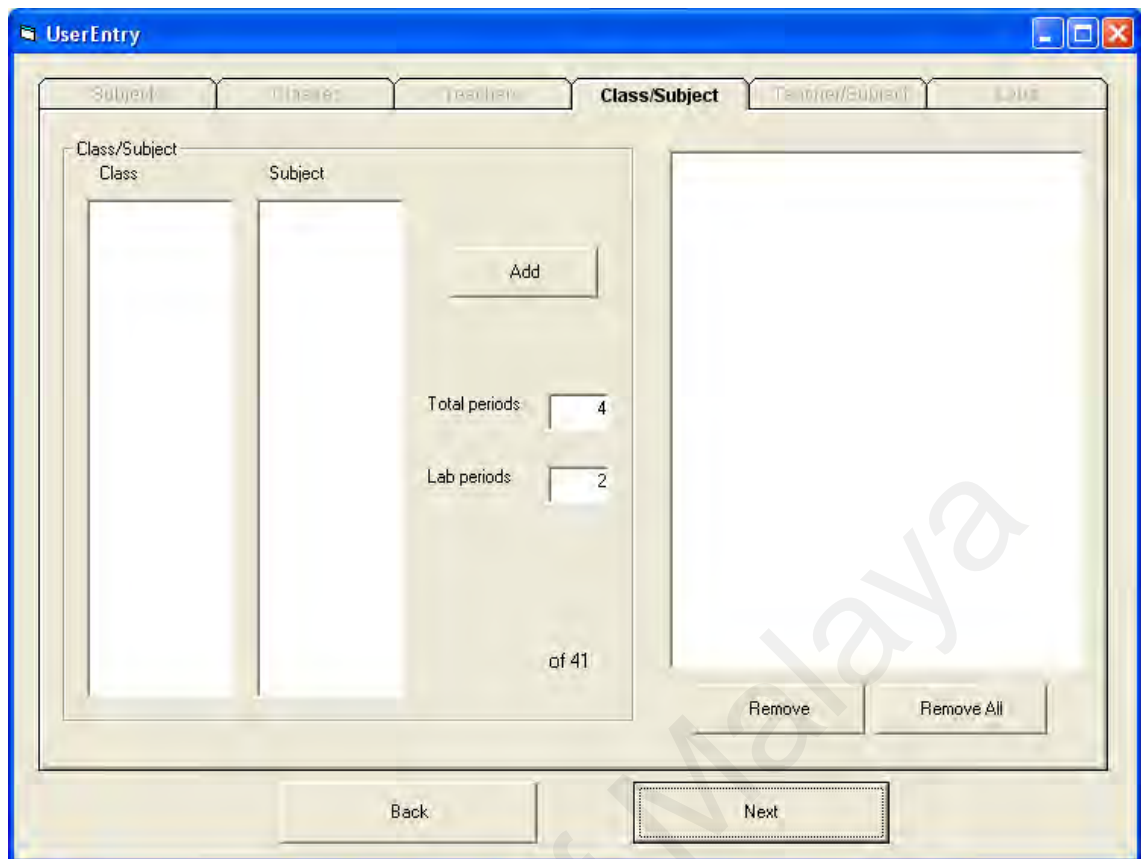


Figure 4.13: Class/Subject Layer.

Layer 5: Teacher/Subject

This layer almost as same as the previous layer the different is that teachers match to subjects they able to teach (see figure 4.14).

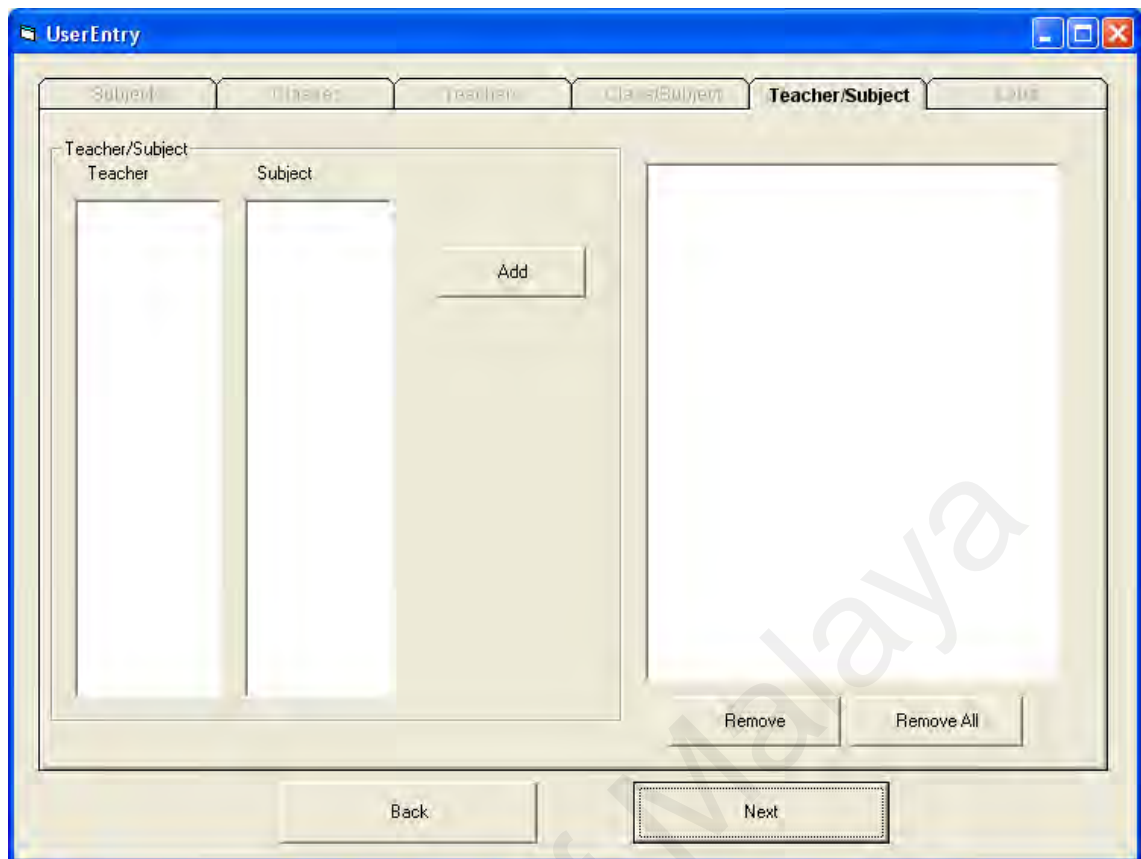


Figure 4.14: Teacher/Subject Layer.

Layer 6: Labs

In this layer user can match list of lessons to specific lab room (see figure 4.15).

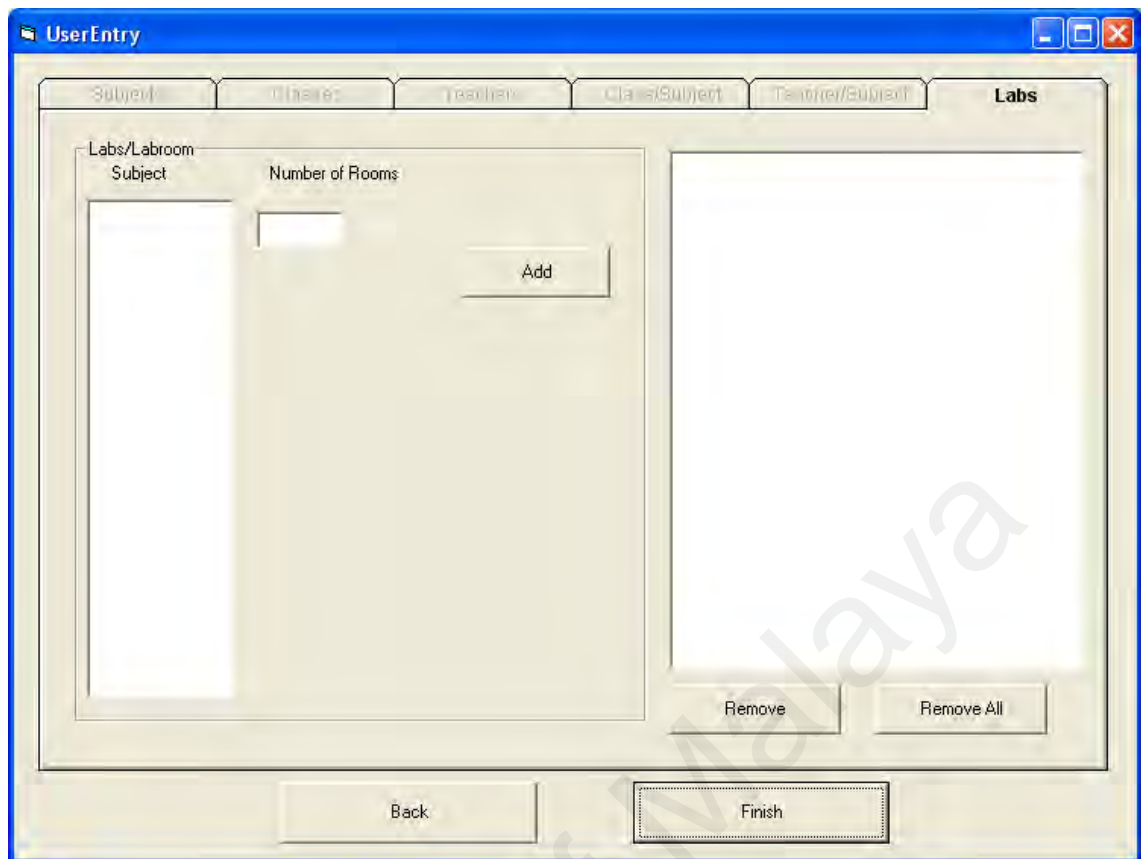


Figure 4.15: Labs Layer.

The most common sharing components list box which is repeated with all the layers to help user reviewing, change, remove, and modify entered data. Add bottoms help user to entered data to the database of system. Each layer provided with two remove and remove all bottom to modify data. The last three bottoms are Back, Next, and Finish bottom. The task of the first and second bottoms is to navigate between layers. Finish bottom if to finalize and save data to main database file presented by text file named “maindata.txt” (see table 4.3 and 4.4).

4.10.2 GA Input/Output User Form

Figure 4.16 represents the main UI of GA. This form allows the user to interact with the GA system by choosing, for example, the crossover method, the selection technique, the stop condition ..., etc. It is also used as the output screen. The main

controls as starting GA and stopping it are represented as command buttons (push buttons), Figure 4.16 is a snapshot of the main form.

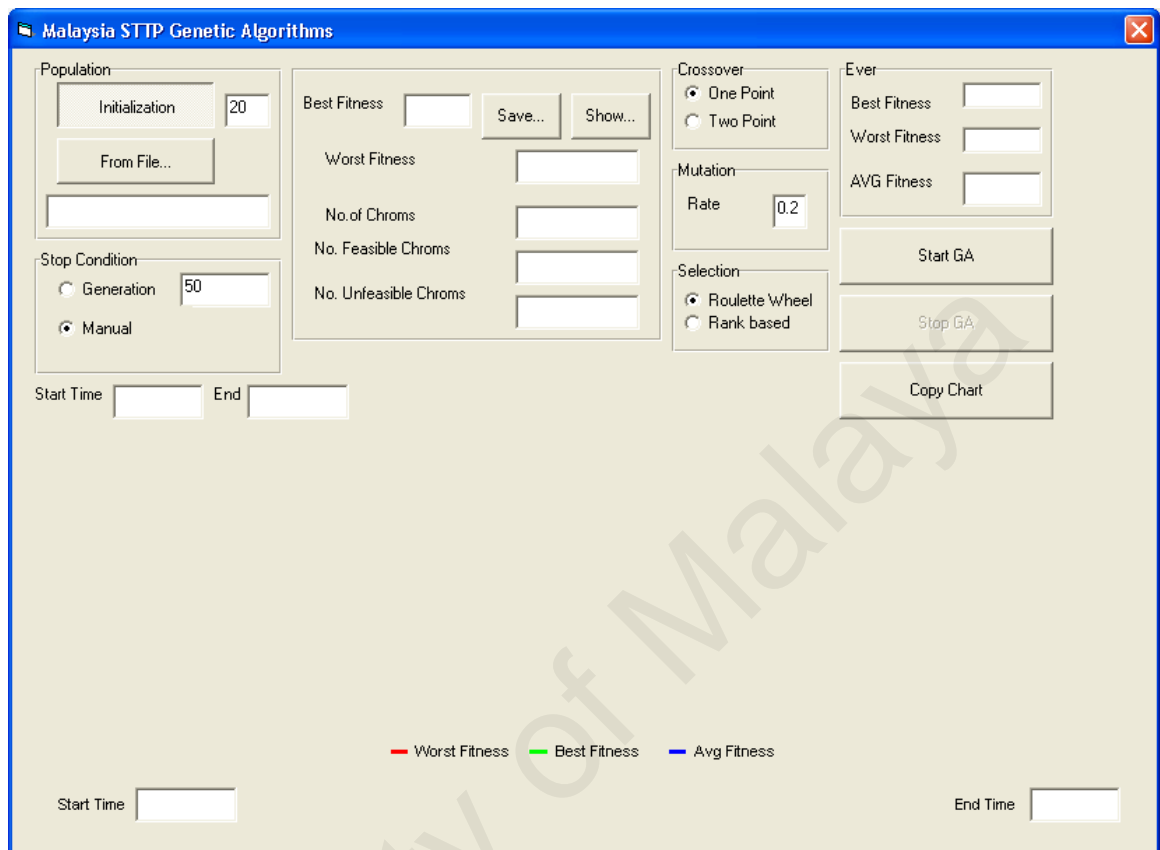


Figure 4.16: Main UI

This form has the following controls:

- “Initialization” and “From File” Toggled Push button

These buttons allow the user to chose whether the population is to be read from file or generated. If the user desired to get the population from a file, the standard Microsoft Visual basic CommonDialog will be activated, that allows the user to choose the file name and location of the input file. (See figure 4.17).

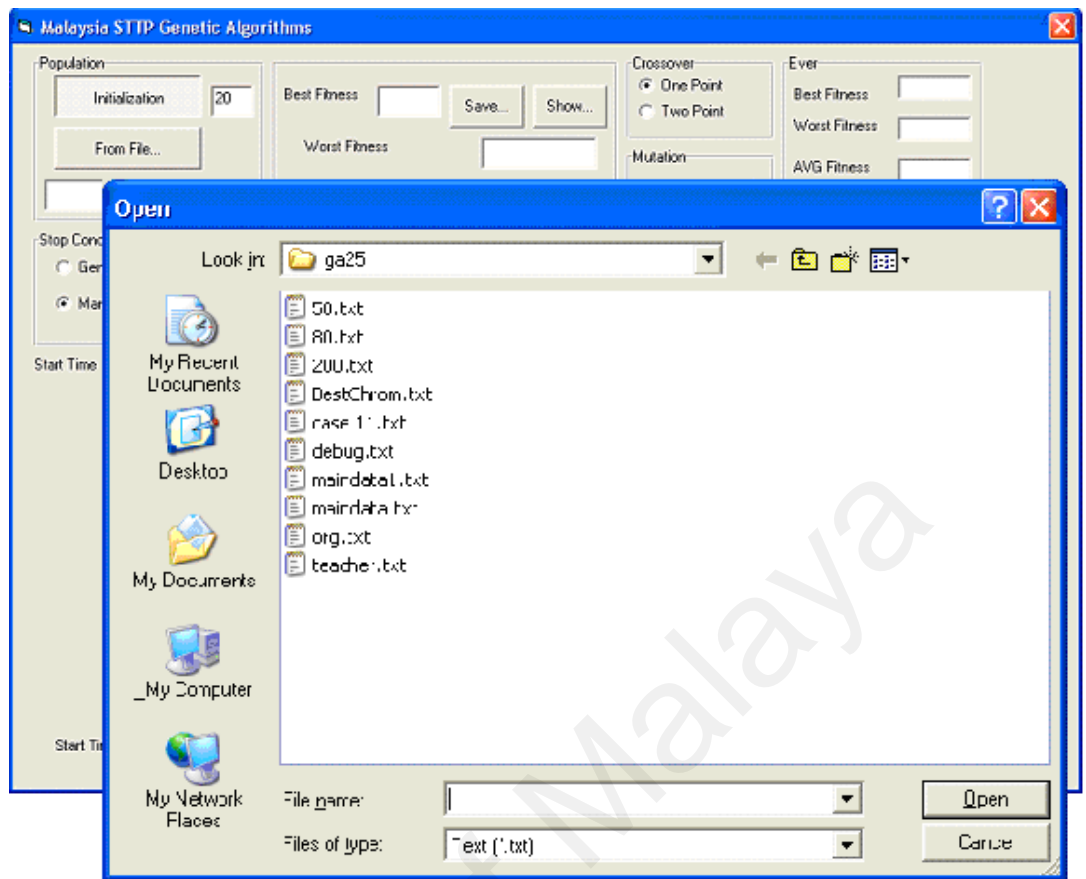


Figure 4.17: User Reads Population from a Text File

The other choice is generating the population randomly. In front of the “Initialization” push button there is Textbox which is used as the number of Chromosomes that needs to be generated (see figure 4.16).

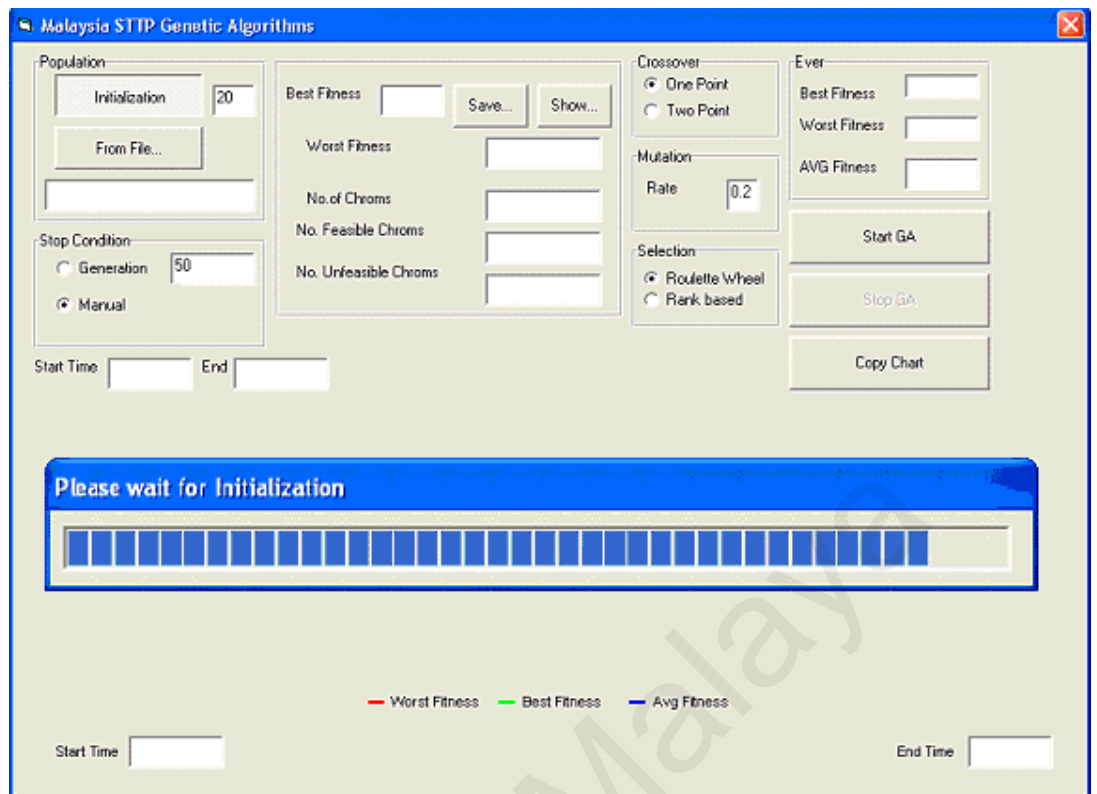


Figure: 4.18 Chromosomes Initialization

Figure 4.18 shows that 20 chromosomes are being created. The progress bar appears to show how the initialization has progressed. Above the progress bar there is two text boxes “Start time” and “End” to show initialization time to obtain the required number of chromosomes.

“Stop Condition” Frame. This frame groups two Radio style option buttons that reflects the two methods which are used to stop the GA iterated process. The textbox in front of the “Generation” button is used to reflect the number of generation/iterations the GA needs to run. The other choice of stopping the GA is manually. If chosen, it will enable in the run-time two command buttons “Resume GA” and “Stop GA” (see figure 4.19).

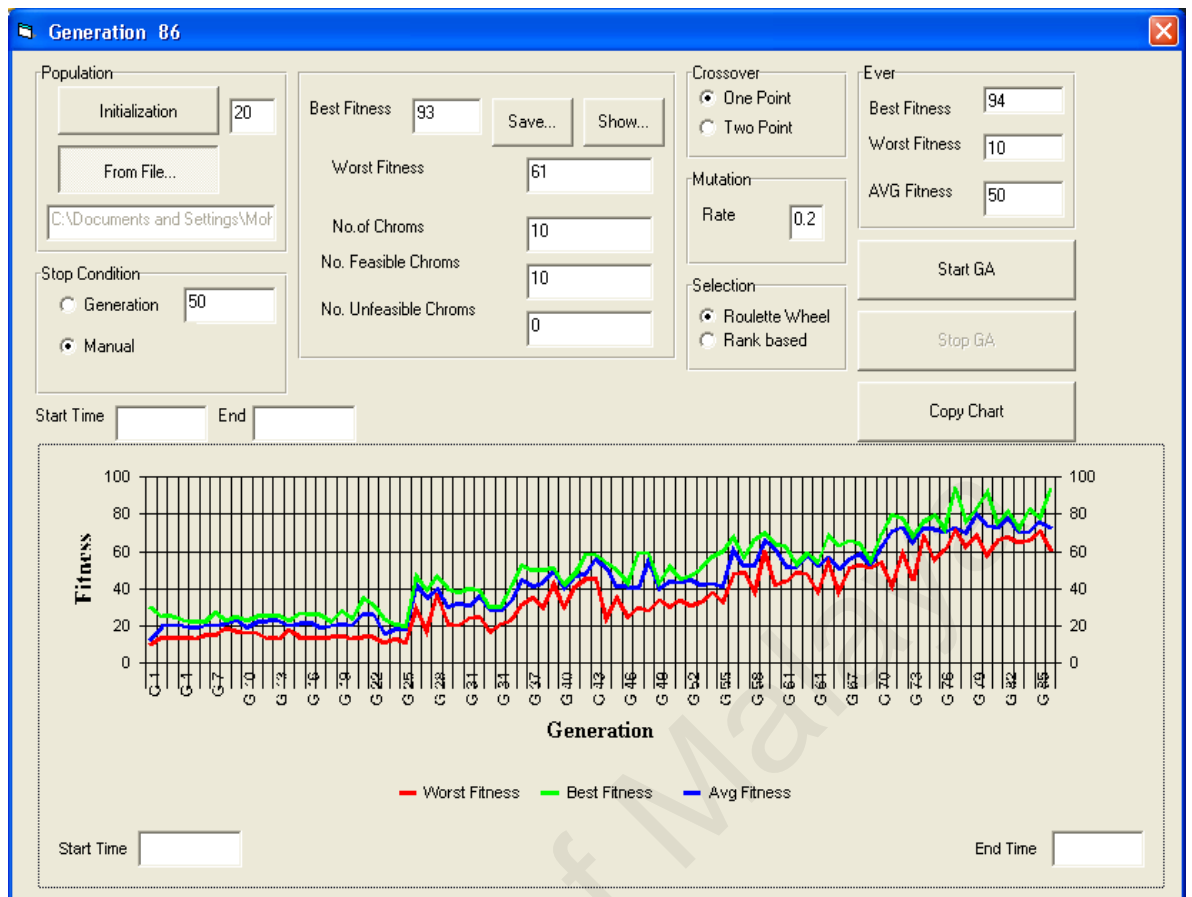


Figure 4.19: GA at Run Time.

Grid Form “Master school timetable form”

The Grid Form is used to show the best, average, and worst chromosomes generated by the GA. At the top right of figure 4.19 there is three text boxes to show the the best, average, and worst fitnesses. About the generation and at the middle of the form you will see the following text boxes:

- Best fitness
- Worst fitness
- Number of chromosomes generated
- Number of feasible chromosomes
- Number of unfeasible chromosomes

These factors represent its values for the current generation and not over all the generations.

At the end of generating chromosomes process user is allowed to press “save” bottom (see figure 4.19) then the standard Microsoft Visual basic CommonDialog will be activated. It allows the user to choose the file name and the location of saving the chromosome in a text file (figure 4.20).

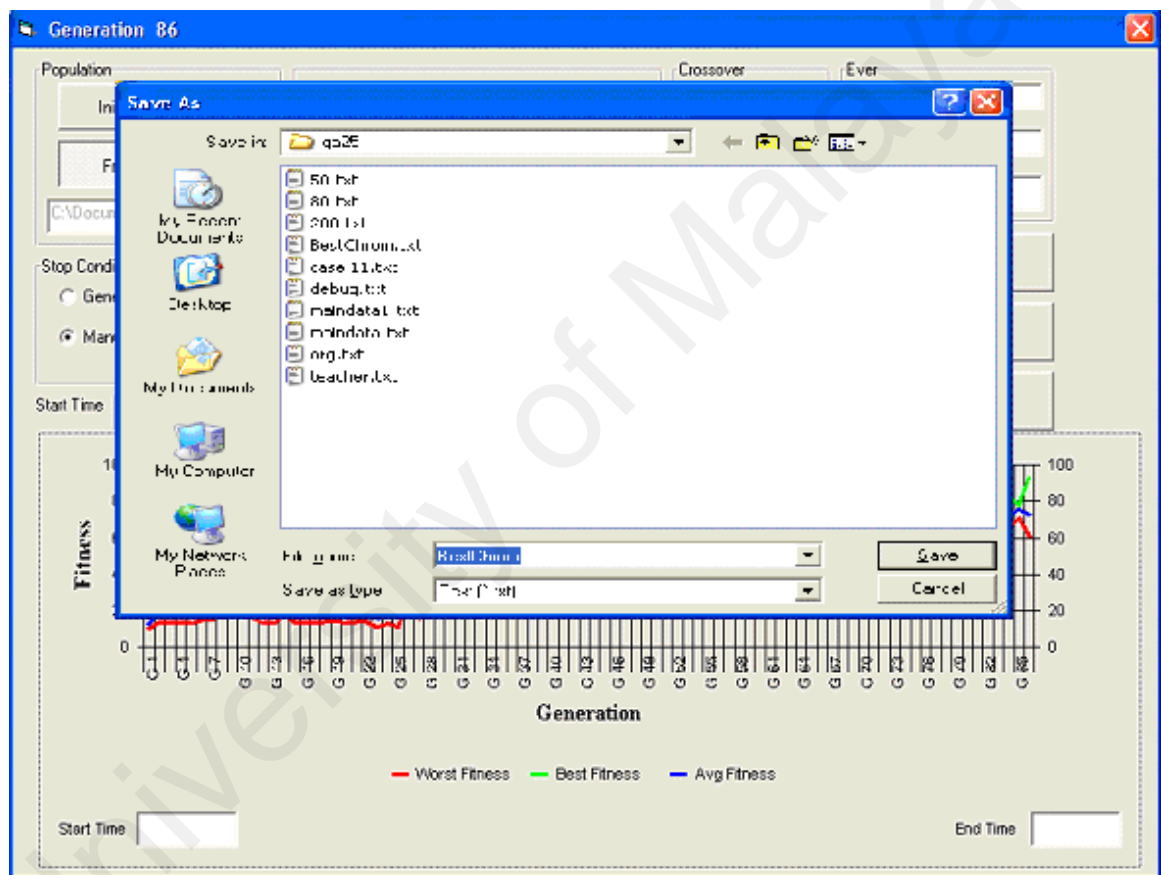


Figure 4.20: Save the Best Chromosome Fitness.

Once the Initialization and generation parts are completed, the “show” bottom at the middle of main form allows the user to review the final result of the system (see figure 4.21).

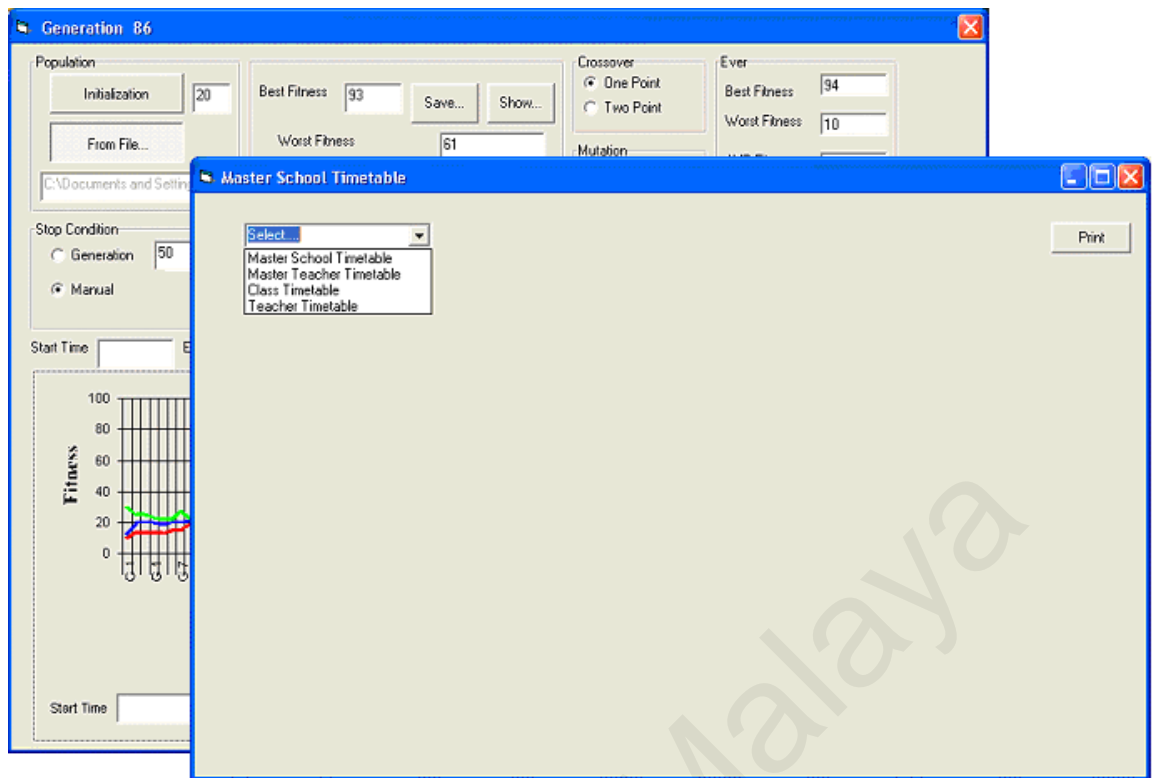


Figure 4.21: GA Result

Figure 4.22 shows the four main components of the system results which are:

- Master school timetable
- Master teacher timetable
- Individual teacher timetable
- Individual class timetable

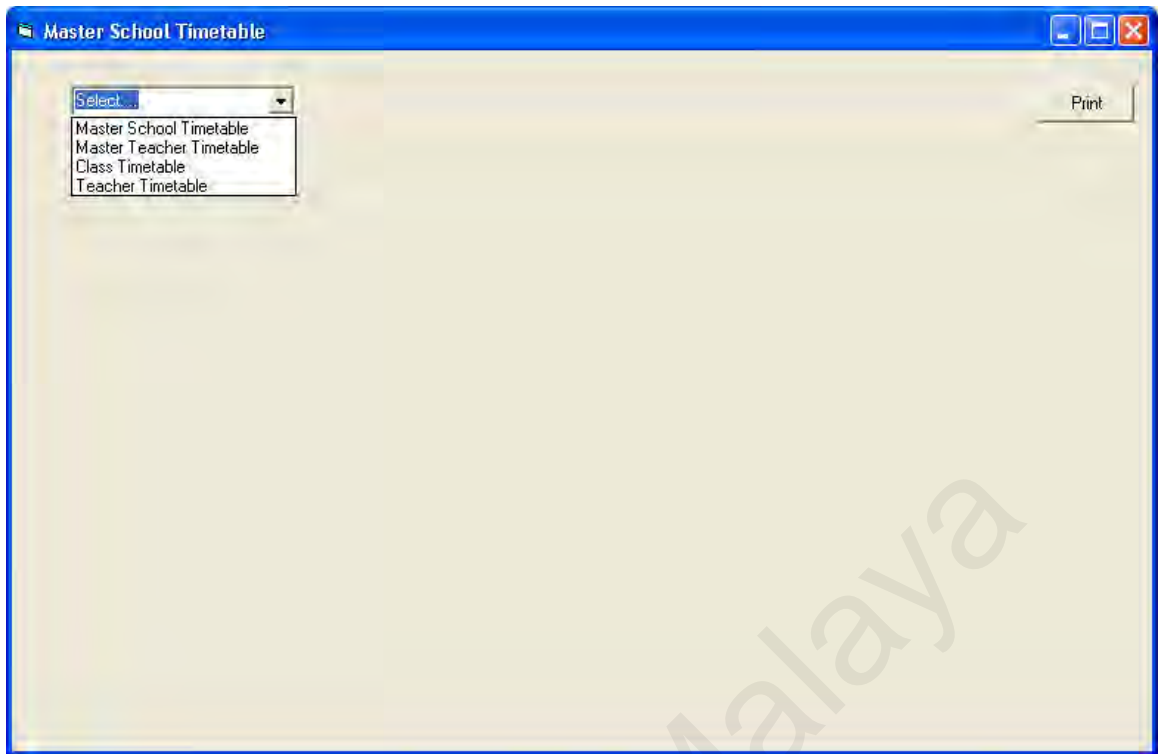


Figure 4.22: List of Timetable

Figure 4.23 represent the master school time table which consists of all the events (lessons) and parameters (teachers). Rows represent classes, columns represent periods and each 9 periods represent one day of the week.

Year	Class	Monday									
		1	2	3	4	5	6	7	8	9	
3	1	BI	5 BM	25 BI	5 GE	29 PJ	7 MA	27 BM	25 BM	25 XX	0
	2	BI	6 SE	27 MA	27 BC	16 MA	27 PM	6 SN	7 SN	7 XX	0
	3	MA	2 GE	29 SE	6 BI	6 BI	6 BM	25 PM	27 PM	27 XX	0
	4	SE	27 PS	5 BM	25 BM	25 PJ	16 SN	7 BI	6 MA	2 XX	0
4	1	EA	31 BM	32 AM	11 BM	32 AM	11 MA	10 AM	11 BM	32 XX	0
	2	PN	15 EA	15 BM	32 MA	24 PM	24 SE	18 SN	21 SN	21 XX	0
	3	BM	14 PJ	16 MA	10 EA	15 MA	10 SN	30 BC	12 PM	24 XX	0
	4	SE	9 PJ	7 AM	13 PN	31 AM	13 BI	11 AM	13 SN	28 XX	0
	5	SN	30 EA	31 MA	24 SN	30 SN	30 EA	31 PS	19 CP	8 XX	0
	6	BI	22 BI	22 AM	14 MA	10 AM	14 BM	23 AM	14 MT	10 XX	0
	7	MA	24 MA	24 PD	17 BI	11 SN	26 SN	26 MA	24 CP	12 XX	0
5	1	BM	23 BI	8 PN	23 BI	8 PD	17 PS	19 BI	8 EA	15 XX	0
	2	AM	19 SE	9 BI	8 PS	5 AM	19 BM	3 MA	28 PD	17 XX	0
	3	SN	21 SN	21 EA	15 BI	22 BM	3 PJ	16 MA	30 PM	29 XX	0
	4	BI	8 SN	26 SN	26 PM	14 BC	12 PJ	29 EA	15 BM	23 XX	0
	5	BM	13 PS	19 SE	18 MA	28 SE	18 MA	28 PM	29 PS	19 XX	0
	6	AM	29 MA	28 MA	28 BM	18 AM	29 BI	22 BM	18 BM	18 XX	0

Figure 4.23: Master School Timetable

The background colors are used to demonstrate different subjects. For instance, the red background color represents Group subjects and the green background color represents Lab classes, the rest are background colored white (see figure 4.23) of Master school timetable.

This form is the same as master school timetable. The only one difference is that the rows represent teachers (t1, t1, ...etc). See Figure 4.24

Master School Timetable

Master Teacher Timetable

Print

Name	No.	Monday																	
		1	2	3	4	5	6	7	8	9	1	2	3	4	5	6	7	8	9
t1	1																		
t2	2	MA	3-3														MA	3-4	
t3	3									BM	5-3	BM	5-2						
t4	4																		
t5	5	BI	3-1	PS	3-4	BI	3-1	PS	5-2										
t6	6	BI	3-2			SE	3-3	BI	3-3	BI	3-3	PM	3-2	BI	3-4				
t7	7			PJ	4-4					PJ	3-1	SN	3-4	SN	3-2	SN	3-2		
t8	8	BI	5-4	BI	5-1	BI	5-2	BI	5-1					BI	5-1	CP	4-5		
t9	9	SE	4-4	SE	5-2														
t10	10					MA	4-3	MA	4-6	MA	4-3	MA	4-1				MT	4-6	
t11	11					AM	4-1	BI	4-7	AM	4-1	BI	4-4	AM	4-1				
t12	12									BC	5-4			BC	4-3	CP	4-7		
t13	13	BM	5-5			AM	4-4			AM	4-4			AM	4-4				
t14	14	BM	4-3			AM	4-6	PM	5-4	AM	4-6			AM	4-6				
t15	15	PN	4-2	EA	4-2	EA	5-3	EA	4-3					EA	5-4	EA	5-1		
t16	16			PJ	4-3			BC	3-2	PJ	3-4	PJ	5-3						
t17	17					PD	4-7			PD	5-1						PD	5-2	
t18	18					SE	5-5	BM	5-6	SE	5-5	SE	4-2	BM	5-6	BM	5-6		
t19	19	AM	5-2	PS	5-5					AM	5-2	PS	5-1	PS	4-5	PS	5-5		

Figure 4.24: Master Teacher Timetable

The last two figures 4.25 and 4.26 show the individual teacher and class timetable. In class timetable the form provided with two ListBox. The first one allows the user to choose year of study and the second specifies the classroom.

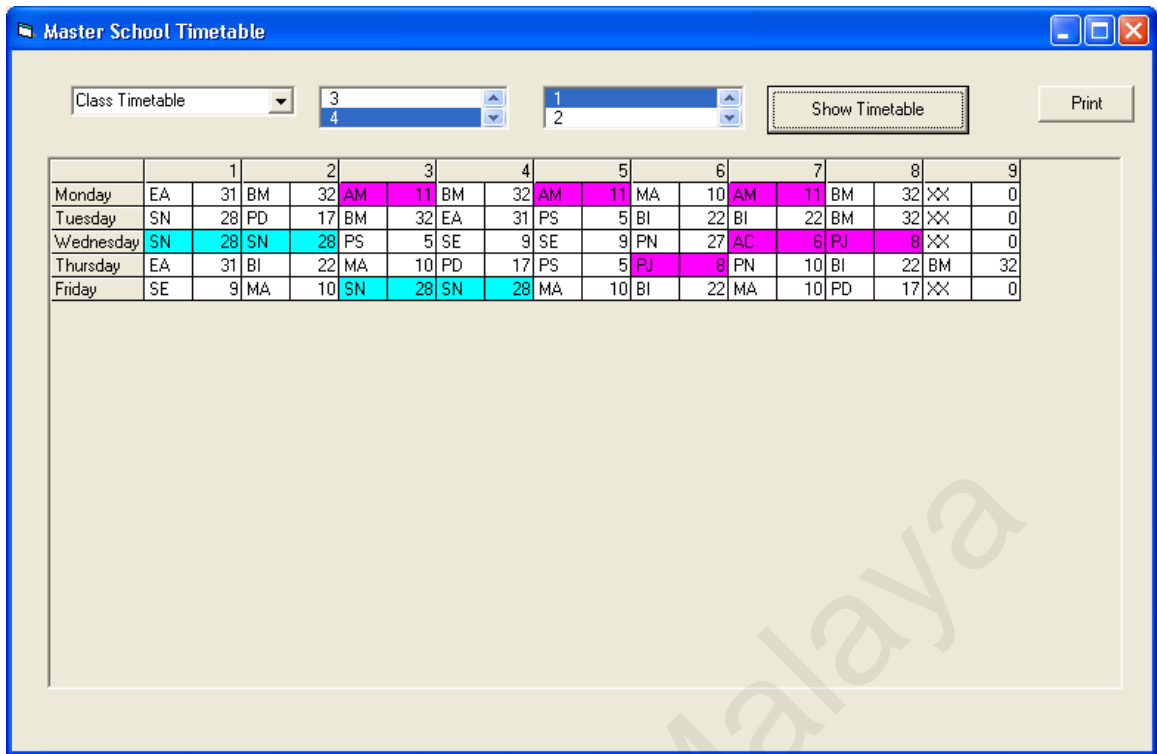


Figure 4.25: Class Timetable

Each teacher can be scheduled individually in the timetable. From figure 4.26, by choosing a teacher from the ListBox and then clicking on “show Timetable” button, all the subjects taught by that teacher will be listed.

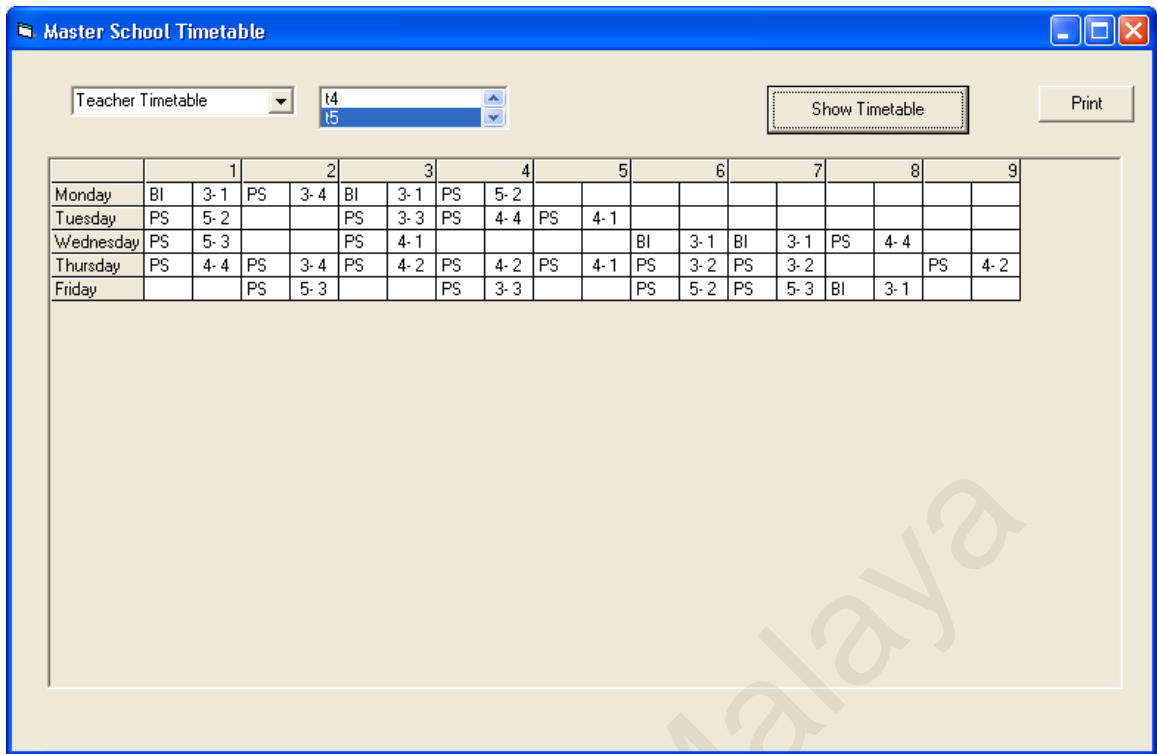


Figure 4.26: Teacher Timetable.

Chapter 5: GA Results and Discussion

5.1 Introduction

In this chapter, the experiment is carried out to solve a real life Malaysia School Timetable Problem using Genetic Algorithms (MSTP-GA). The example experiment, and the result or output of the system will be outlined. Issues such as population size, generation iteration times, time of initialization and generation to obtain optimal or near to optimal solution will be discussed. Operators used in these experiments were of the normal type crossover (one and two point crossover), normal mutation, and selection techniques (Rank based selection and Roulette wheel) with a variety number of GA iterations.

5.2 Programming Language and System Requirement

The experiment was tested on a computer with the following attributes:

OS	Microsoft Windows XP Professional
Version	5.1.2600 Service Pack 1 Build 2600
Processor	598 Mhz, pentium III
Hard disk	20 GB
Physical Memory	256.00 MB
Virtual Memory	874.46 MB

The Visual Basic 6 and Visual studio in windows XP environment is used to develop this system.

5.3 Data Formatting

The real data is collected from a number of Malaysian secondary schools. Some of them are listed below (see table 5.1). The technique used for gathering the data is a questionnaire (see Appendix C).

Table 5.1: Malaysian Secondary Schools.

No	Name of school
1	Sek Bandar Baru Seri Petaling, Kuala Lumpur
2	Smk Damansare Jaya
3	S. M. K. (L) Bukit Bintang
4	Smk (P) Methodist, K.L.
5	Smk Taman Kosas. 68000 Ampang, SgZangoe
6	S.M.K L Salle, Petaling Jaya
7	Smk Petaling, Kuala Lumpur
8	Sek. Keb. Petaling 2, Kuala Lumpur
9	Sekolah KebangsaanTaman Kosas
10	Sekolah Menengah (Per) Pudu
11	Sek Seksyen 11, Shah Alam

Most of these schools are still developing their system manually. Some are using applications like Microsoft excel and word to manage their timetable problem and few of them only are using dedicated software systems for timetabling and even those dedicated software systems do not really produce the timetable that really satisfies students and staff wishes. For example, lab lessons are preferred to be two continual periods which are difficult to manage over the entire timetable for each class and lab lesson. In simple words it is difficult to produce a timetable that satisfies most of the school preferability without using optimization techniques like Genetic Algorithms.

Schools are different from universities in that students in school have one session a year scheduled on a weekly basis whereas in university group of students are scheduled to one class and almost have the same lessons, but some lessons must be taught for a group of students from different classes (usually two classes). Moreover, the data is arranged in rows class by class. For example, class 1/1 means students who are in first year are put in room 1 class 1/2, 2/1 and so on. Days and periods are arranged as columns (see Figure 5.2). In this project the data is grouped into one text file. Figure 5.1a, 5.1b and 5.1c show how the data is grouped into the file.

teacher	Subject				
T1	AC				
T2	MA				
T3	BM				
T4	AC	AM			
T5	BI	PS			
T6	BI	SE	AM	PM	AC
T7	SN	PJ			
T8	BI	PJ	CP		
T9	SE	AC	BC	CP	
t10	MA	PN	MT		
t11	BI	AM	CP	PS	
t12	PM	AC	BC	CP	GE
t13	KH	AM	BM		
t14	BM	AM	AC	PM	
t15	PN	EA			
t16	PX	BC	PN		
t17	PD	PN			
t18	PN	SE	BM		
t19	PS	AM	PN		
t20	KH				
t21	SN	FZ	PN		
t22	BI	AC			
t23	BM	PN			
t24	MA	PM			
t25	BM	CP			
t26	SN	PN	KM		
t27	MA	AM	PM	SE	PN
t28	MA	SN	AC		
t29	GE	PM	PJ	AM	PN
t30	BG	MA	SN	PN	
t31	EA	PN	PA		
t32	SE	BM			

Figure 5.1a: Relationship between Teachers and Subject

Lab	No of room
SN	2
BG	3
FZ	2
KM	3
PJ	1

Figure 5.1 b: Subjects and Their Respective Laboratories

	MA	BI	SN	KH	PN	BG	EA	PD	PS	BM	PM	GE	SE	AC	AM	PJ	PA	CP	MT
c1	5	5	5	4	0	0	0	0	2	6	0	3	3	3	3	2	0	0	0
c2	5	5	5	4	0	0	0	0	2	6	3	3	3	0	0	2	0	0	0
c3	5	5	5	4	0	0	0	0	2	6	3	3	3	0	0	2	0	0	0
c4	5	5	5	4	0	0	0	0	2	6	0	3	3	3	3	2	0	0	0
c5	5	5	5	0	2	0	3	3	3	6	0	0	3	1	3	2	0	0	0
c6	5	5	5	0	3	0	3	3	3	6	3	0	3	0	0	2	0	0	0
c7	5	5	5	0	0	0	3	0	3	6	3	0	3	0	0	2	4	0	0
c8	5	5	5	0	1	0	3	0	3	6	0	0	3	1	3	2	4	0	0
c9	5	5	5	0	0	0	3	3	3	6	3	0	3	0	0	2	0	3	0
c10	5	5	0	0	0	4	0	0	0	6	0	0	3	1	3	0	0	0	4
c11	5	5	5	0	0	0	3	3	3	6	3	0	3	0	0	2	0	3	0
c12	5	5	5	0	3	0	3	3	3	6	3	0	3	0	0	2	0	0	0
c13	5	5	5	0	0	0	0	3	3	6	0	3	3	1	3	2	0	2	0
c14	5	5	5	0	2	0	3	0	3	6	3	0	3	0	0	2	4	0	0
c15	5	5	5	0	0	0	3	0	3	6	3	0	3	0	0	2	4	0	0
c16	5	5	5	0	0	0	3	3	3	6	3	0	3	1	0	2	0	2	0
c17	5	5	5	0	2	0	3	3	3	6	0	0	3	1	3	2	0	0	0

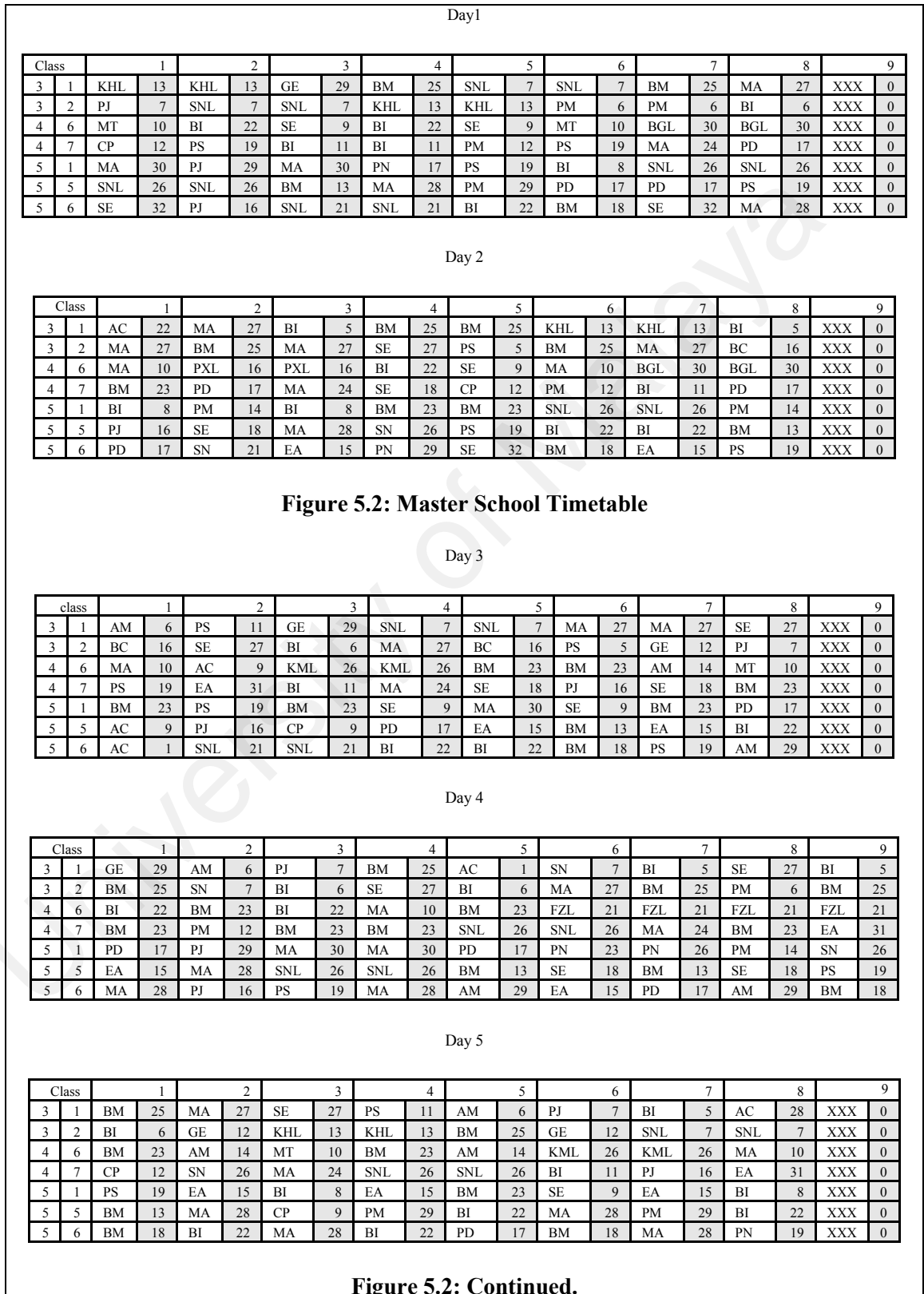
Figure 5.1c: Relationship Between Class and Number of Lessons per Week

Figure 5.1c shows the subjects, referred to by short name such as MA, BI ..etc. the notations c1, c2, ..etc refer to the classes (i.e., year of study and its room) . The numbers distributed over the entire table refer to number of lessons of subject that must be taught for each class per week. For example, subject MA is 5 lessons for each class per week.

The following is the steps involve in the processing the data using visual basic:

1. Read the input file from text file “maindata.txt” (see figures 5.1a, 5.1b and 5.1c).
2. Use the combination of arrays and simple link structure for the programming part. It is divided into three main parts:
 - Part 1: read all teachers based on their subjects which they able to teach, then check the duplication of teachers. If no duplication found, add teacher to the list. Once done, skip to another teacher.
 - Part 2: the purpose of this part is to match a teacher to a particular class based on what subject the class has.

- Part 3: write the data structure into output file. The format of the output file is; each row represents specific class, each column represents period and each 9 periods represent one day (see Figure 5.2).



The above table is the result of this system and the only modification done on the table is that the days are reordered in columns instead of rows (i.e., days followed each other).

5.4 Experiments

Real data is used with these experiments (see figure 5.1). The final result can be either a master school timetable (see figure 5.2), a master teacher timetable, a individual teacher timetable or a individual class timetable. This chapter experiments focus on the best fitness that could be obtained with fixed population size and different generation iterations. The result of implementing this experiment will be represented by curves. Each curve has two axes X and Y. X represents fitness values and Y represents generation iterations.

The experiments are tested with 697 events in 17 classrooms and 32 teachers. Events are lessons, labs and groups. Events are based on data collected for different Malaysian secondary schools. The penalty function and set of constraints are defined in Chapter 3. All the chromosomes passed to GA are feasible solution all the time.

The user determines the number of chromosomes that needs to be initialized. These chromosomes are members of first generation. These chromosomes will be saved by the system to a text file called "debug.txt" (see figure 5.3). Then these members will be used to produce a child for the next generation. User can choose one of the operations to produce the child rate of mutation, type of crossover and parents selection techniques. The number of generations is decided by the user also, which will be 20, 40, 60 and 100 in this experiment to find out in what generation the fittest chromosome can be obtained.

```

debug.txt - Notepad
File Edit Format View Help
60
3,1,"SNL",7,"SNL",7,"AC",22,"GE",29,"PJ",7,"MA",27,"AM",6,"AC",28,"XXX",0,"GE",29,"AM",6,"AM",6,"BM",25,"PJ",7,"KHL",13,"
3,2,"MA",27,"PM",6,"BI",6,"PJ",7,"MA",27,"BC",16,"SN",7,"BI",6,"XXX",0,"BM",25,"GE",12,"GE",12,"SE",27,"MA",27,"F
3,3,"BM",25,"KHL",13,"KHL",13,"PJ",16,"BI",6,"PS",5,"BC",12,"GE",29,"XXX",0,"SE",6,"SNL",7,"SNL",7,"SE",6,"MA",2,"SE",6,"
3,4,"MA",2,"BM",25,"AC",12,"BM",25,"PJ",16,"BI",6,"AM",27,"AC",12,"XXX",0,"PS",5,"AM",27,"AM",27,"SE",27,"PJ",16,"SN",7,"
4,1,"BI",22,"MA",10,"BM",32,"EA",31,"BM",32,"BM",32,"BI",22,"PN",10,"XXX",0,"BM",32,"BM",32,"PS",5,"MA",10,"PS",5,"PJ",8,"
4,2,"PM",24,"EA",15,"PS",5,"PD",17,"PN",21,"BI",11,"BI",11,"BI",11,"MA",24,"XXX",0,"SE",18,"BI",22,"BI",11,"MA",24,"MA",10,"
4,3,"PA",31,"PA",31,"SE",18,"SE",18,"BI",8,"PS",19,"PA",31,"BM",14,"XXX",0,"BI",8,"EA",15,"PS",19,"PM",24,"SE",18,"MA",10,"
4,4,"SE",9,"BM",32,"MA",24,"BI",11,"PS",5,"MA",24,"PS",5,"BM",32,"XXX",0,"MA",24,"PA",31,"MA",24,"PS",5,"BI",11,"EA",15,"
4,5,"BI",11,"PS",19,"BI",11,"MA",24,"PM",14,"PD",17,"MA",24,"PS",19,"XXX",0,"BM",14,"BM",14,"SNL",30,"SNL",30,"SNL",30,"
4,6,"MA",10,"BGL",30,"BGL",30,"SE",9,"BI",22,"MA",10,"MT",10,"BM",23,"XXX",0,"BM",23,"BM",23,"BI",22,"FZL",21,"FZ
4,7,"PM",23,"BM",23,"BM",23,"PM",12,"MA",24,"SNL",26,"SNL",26,"EA",31,"XXX",0,"PS",19,"SE",18,"SE",18,"SNL",26,"SNL",26,"
5,1,"PM",14,"SE",9,"PD",17,"PN",23,"EA",15,"EA",15,"BI",8,"PN",26,"XXX",0,"MA",30,"MA",30,"PN",17,"PS",19,"PM",14,"PM",14,"
5,2,"PS",5,"PD",17,"SNL",21,"SNL",21,"AM",19,"MA",28,"CP",25,"BM",3,"XXX",0,"PD",17,"SE",9,"CP",25,"AC",14,"GE",29,"MA",28,"
5,3,"MA",30,"SNL",21,"PM",29,"BI",22,"PA",31,"PN",18,"PJ",16,"MA",30,"XXX",0,"EA",15,"BI",22,"BM",3,"PA",31,"BI",22,"PS",5,"
5,4,"EA",15,"SNL",26,"SNL",26,"SNL",26,"SNL",26,"BC",12,"PJ",29,"BI",8,"XXX",0,"MA",10,"BI",8,"BI",8,"BM",23,"PS",19,"PS",19,"
5,5,"BM",13,"SE",18,"MA",28,"BM",13,"SE",18,"PM",29,"CP",9,"BI",22,"XXX",0,"SNL",26,"SNL",26,"CP",9,"AC",9,"MA",28,"PD",17,"
5,6,"PD",17,"MA",28,"PS",19,"PS",19,"AM",29,"SNL",21,"SNL",21,"BM",18,"XXX",0,"MA",28,"MA",28,"SE",32,"AC",1,"SE",32,"BM",
3,1,"PS",11,"MA",27,"AM",19,"SNL",7,"SNL",7,"BI",5,"MA",27,"BM",25,"XXX",0,"BM",25,"BM",25,"BM",25,"MA",27,"AM",6,"GE",29,"
3,2,"BC",16,"SN",7,"BC",16,"BC",16,"SE",27,"KHL",13,"KHL",13,"MA",27,"XXX",0,"MA",27,"SE",27,"PM",6,"BM",25,"PJ",7,"PS",19,"
3,3,"PM",27,"BC",12,"BC",12,"BM",25,"BM",25,"PS",5,"MA",2,"XXX",0,"GE",29,"BI",6,"KHL",13,"KHL",13,"PJ",16,"KHL",13,"
3,4,"BI",6,"BM",25,"AM",27,"KHL",13,"KHL",13,"GE",12,"SNL",7,"SNL",7,"XXX",0,"PS",5,"MA",2,"PS",5,"MA",2,"AM",27,"SE",27,"
4,1,"BM",32,"BM",32,"PS",5,"MA",10,"PJ",8,"BI",22,"BI",22,"BM",32,"XXX",0,"MA",10,"SE",9,"SNL",28,"SNL",28,"BI",22,"MA",10,"
4,2,"EA",15,"EA",15,"PD",17,"MA",24,"MA",24,"BI",11,"SN",21,"PN",16,"XXX",0,"PM",24,"BI",11,"PJ",8,"BI",11,"PS",5,"SE",18,"
4,3,"BI",8,"PS",19,"BM",14,"SN",30,"MA",10,"PM",24,"BM",14,"PM",24,"XXX",0,"BI",8,"PA",31,"BC",12,"MA",10,"PM",24,"BI",8,"
4,4,"SNL",28,"SNL",28,"EA",15,"EA",15,"PA",31,"EA",15,"BM",32,"SE",9,"XXX",0,"PA",31,"PS",5,"BI",11,"PA",31,"BI",11,"BI",11,"
4,5,"SE",18,"BI",11,"SE",18,"BM",14,"BI",11,"PM",14,"PD",17,"BI",11,"XXX",0,"PM",14,"PS",19,"PJ",16,"MA",24,"EA",31,"EA",31,"
4,6,"FZL",21,"FZL",21,"MT",10,"BM",23,"BGL",30,"BGL",30,"SE",9,"MA",10,"XXX",0,"BGL",30,"BGL",30,"MT",10,"SE",9,"KML",26,"
4,7,"PM",12,"MA",24,"BI",11,"BI",11,"PJ",16,"PS",19,"BM",23,"BM",23,"XXX",0,"SE",18,"BM",23,"PD",17,"BM",23,"PD",17,"MA",
5,1,"BM",23,"PJ",29,"SE",9,"BI",8,"BM",23,"BM",23,"PS",19,"PM",14,"XXX",0,"PD",17,"SNL",26,"SNL",26,"PS",19,"EA",15,"MA",
5,2,"BM",3,"PS",5,"AM",19,"GE",29,"PD",17,"BM",3,"BI",8,"SN",21,"XXX",0,"SNL",21,"SNL",21,"SE",9,"BI",8,"CP",25,"MA",28,"
5,3,"PM",29,"PA",31,"PA",31,"PS",5,"PM",29,"PJ",16,"PA",31,"MA",30,"XXX",0,"BI",22,"BI",22,"EA",15,"SNL",21,"SNL",21,"EA",
5,4,"SNL",26,"SNL",26,"BI",8,"SN",26,"PS",19,"PJ",29,"BC",12,"BC",12,"XXX",0,"BM",23,"MA",10,"PA",31,"PM",14,"MA",10,"SE",
5,5,"BM",13,"SE",18,"MA",28,"PS",19,"EA",15,"SNL",26,"SNL",26,"BM",13,"XXX",0,"MA",28,"BM",13,"PM",29,"EA",15,"CP",9,"BI",
5,6,"BI",22,"PJ",16,"AM",29,"SNL",21,"SNL",21,"SN",21,"MA",28,"BI",22,"XXX",0,"SE",32,"PD",17,"PM",18,"BM",18,"BM",18,"SE",
3,1,"GE",29,"PJ",7,"SN",7,"BM",25,"AM",6,"GE",29,"PJ",7,"MA",27,"XXX",0,"AM",6,"GE",29,"SE",27,"KHL",13,"KHL",13,"BM",25,"
3,2,"SN",7,"GE",12,"BM",25,"MA",27,"BM",25,"PJ",7,"MA",27,"BI",6,"XXX",0,"BC",16,"KHL",13,"KHL",13,"SNL",7,"SNL",7,"SE",12,"
3,3,"SE",6,"BI",6,"BI",6,"GE",29,"PS",5,"PJ",16,"BM",25,"MA",2,"XXX",0,"BM",25,"SNL",7,"SNL",7,"BM",25,"MA",2,"SNL",7,"SNL",7,"
3,4,"BM",25,"PJ",16,"GE",12,"SN",7,"AM",27,"BM",25,"PJ",16,"GE",12,"XXX",0,"AM",27,"BM",25,"MA",2,"BI",6,"BI",6,"MA",2,"E
4,1,"MA",10,"BM",32,"PN",27,"SN",28,"EA",31,"PS",5,"MA",10,"BI",22,"XXX",0,"EA",31,"SE",9,"BM",32,"BM",32,"AM",11,"
4,2,"EA",15,"BM",14,"MA",10,"SNL",30,"SNL",30,"BI",8,"BM",14,"PJ",16,"XXX",0,"BI",11,"BM",32,"PM",24,"PD",17,"MA",24,"EA",11,"
4,3,"PA",31,"BM",14,"MA",10,"SNL",30,"SNL",30,"BI",8,"BM",14,"PJ",16,"XXX",0,"BI",8,"BM",14,"BI",8,"MA",10,"SNL",30,"SNL",
4,4,"PA",31,"SNL",28,"MA",24,"PM",32,"MA",24,"BM",32,"PJ",7,"XXX",0,"BM",32,"PS",5,"BI",11,"PA",31,"BI",11,"AM",
4,5,"BM",14,"EA",31,"BI",11,"BI",11,"PD",17,"CP",11,"PS",19,"PD",17,"XXX",0,"MA",10,"EA",31,"PS",19,"BM",14,"BM",14,"PS",
4,6,"BI",22,"MA",10,"PKL",16,"PKL",16,"PM",23,"SE",9,"BM",23,"MA",10,"XXX",0,"MA",10,"KML",26,"KML",26,"KML",26,"KML",26,"
4,7,"PS",19,"BM",23,"SNL",26,"SNL",26,"PS",19,"CP",12,"SE",18,"SN",26,"XXX",0,"BM",23,"MA",24,"BM",23,"BI",11,"PM",12,"SE",
5,1,"BM",23,"PN",26,"MA",30,"BM",23,"SE",9,"BM",23,"PD",17,"MA",30,"XXX",0,"MA",30,"BM",23,"MA",30,"BI",8,"SE",9,"BI",8,"
5,2,"PS",5,"CP",25,"GE",29,"SN",21,"MA",28,"PD",17,"SE",9,"AC",14,"XXX",0,"PD",17,"SNL",21,"SNL",21,"AM",19,"MA",28,"BM",

```

Figure 5.3: Number of Chromosome Initialized to be First Members of Generation.

After generating the required numbers, the system will choose the fittest member among members of generations and save it to a text file called “BestChrom.txt”, see figure 5.4 and rearrange the genes and the periods to present a meaningful output. The output is the weekly school timetable, see section 5.4 in figure 5.2.

```

B,1,"MA",27,"PJ",7,"MA",27,"BM",25,"SE",27,"SNL",7,"SNL",7,"BM",25,"XXX",0,"MA",27,"SE",27,"KHL",13,"KHL",13,"MA",27,"BM",
3,2,"BI",6,"MA",27,"SN",7,"KHL",13,"KHL",13,"KHL",13,"KHL",13,"SE",27,"XXX",0,"BM",25,"BI",6,"PM",6,"SE",27,"SNL",7,"SNL",
3,3,"GE",29,"BI",6,"PS",5,"PM",27,"BC",12,"SE",6,"SE",6,"BC",12,"XXX",0,"MA",2,"SNL",7,"SNL",7,"SN",7,"GE",29,"PM",27,"P",
3,4,"MA",2,"PJ",16,"GE",12,"MA",2,"BM",25,"GE",12,"BM",25,"PS",5,"XXX",0,"KHL",13,"KHL",13,"BM",25,"PS",5,"MA",2,"KHL",13,
4,1,"BI",22,"PS",5,"BM",32,"SE",9,"BI",22,"BI",22,"SNL",28,"SNL",28,"XXX",0,"BI",22,"PS",5,"PJ",8,"EA",31,"EA",31,"PJ",8,
4,2,"MA",24,"EA",15,"BI",11,"SNL",21,"SNL",21,"MA",24,"PS",5,"PM",24,"XXX",0,"PD",17,"BM",32,"MA",24,"BM",32,"PN",15,"PD",
4,3,"BM",14,"BM",14,"PM",24,"EA",15,"MA",10,"BI",8,"PA",31,"EA",15,"XXX",0,"BM",14,"PM",24,"BM",14,"EA",15,"MA",10,"SE",1,
4,4,"SE",9,"BM",32,"PA",31,"PS",5,"SN",28,"EA",15,"SE",9,"BM",32,"XXX",0,"MA",24,"BI",11,"PN",31,"SNL",28,"SNL",28,"MA",2,
4,5,"CP",11,"CP",8,"PM",14,"CP",8,"PM",14,"BI",11,"SE",18,"BI",11,"XXX",0,"SE",18,"BM",14,"PD",17,"PM",14,"BM",14,"BI",11,
4,6,"MA",10,"BI",22,"PXL",16,"PXL",16,"BM",23,"MA",10,"BGL",30,"BGL",30,"XXX",0,"MT",10,"MT",10,"MT",10,"FZL",21,"FZL",21,
4,7,"CP",12,"CP",12,"SE",18,"CP",12,"SNL",26,"SNL",26,"BI",11,"PS",19,"XXX",0,"EA",31,"SE",18,"PJ",16,"MA",24,"PS",19,"P",
5,1,"EA",15,"PN",17,"EA",15,"SN",26,"PS",19,"PJ",29,"BM",23,"BM",23,"XXX",0,"EA",15,"PD",17,"MA",30,"BI",8,"BM",23,"SNL",
5,2,"PD",17,"AM",19,"GE",29,"BM",3,"PS",5,"MA",28,"BM",3,"BM",3,"XXX",0,"BI",8,"PJ",29,"SE",9,"CP",25,"BM",3,"SN",21,"BM",
5,3,"MA",30,"MA",30,"BI",22,"SE",18,"PN",30,"PS",5,"PM",29,"SN",21,"XXX",0,"MA",30,"BI",22,"EA",15,"SE",18,"PS",5,"BM",3,
5,4,"SNL",26,"SNL",26,"MA",10,"PA",31,"PA",31,"BM",23,"MA",10,"MA",10,"XXX",0,"BC",12,"EA",15,"SNL",26,"SNL",26,"SE",9,"E
5,5,"BM",13,"MA",28,"BM",13,"PS",19,"PM",29,"PS",19,"PD",17,"BI",22,"XXX",0,"MA",28,"PJ",16,"PM",29,"CP",9,"PD",17,"BI",2,
5,6,"SN",21,"AM",29,"MA",28,"BI",22,"PD",17,"PJ",16,"BI",22,"BM",18,"XXX",0,"SNL",21,"SNL",21,"SE",32,"PS",19,"BM",18,"SE

```

Figure 5.4: Fittest Chromosome

5.4.1 A Case Study

In this case study, 697 events in 17 classrooms and 32 teachers are tested. 100 chromosomes will be initialized to be the first generation members. These elements will be tested with different generations to come out with the fittest chromosome. 20, 40, 60, and 100 generations will be used in this case study. The initialization population started at 2:24:11 PM and finished last the chromosome at 3:06:42 (roughly 42:30 mints). The time of initialization depends on the size of population.

5.5 Case Study Result

5.5.1 Test 1: population size = 100, generation number = 20 .

The chart below shows the performance of the MSTTP-GA on test 1(see figure 5.5). The best chromosome implemented 20% of the soft constraints with 20 generations (see table 5.2).

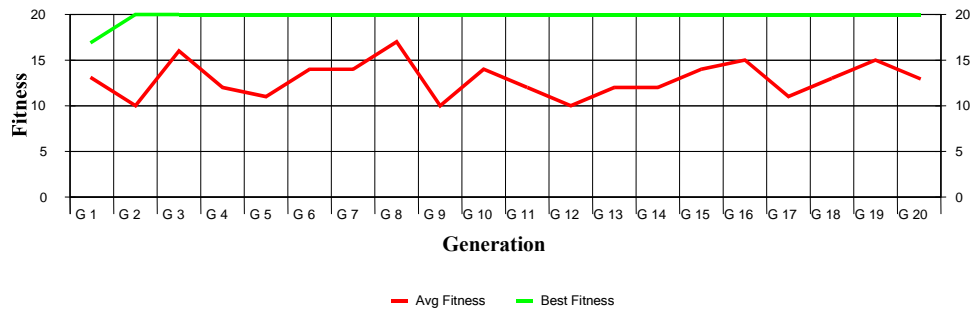


Figure 5.5: Case Study Test 1 Result

Table 5.2: Case Study Test 1 Result

Generation	Best	Avg	VPC
G 1	17	13	9
G 2	20	10	9
G 3	20	16	10
G 4	20	12	9
G 5	20	11	9
G 6	20	14	9
G 7	20	14	10
G 8	20	17	10
G 9	20	10	11
G 10	20	14	11
G 11	20	12	11
G 12	20	10	9
G 13	20	12	10
G 14	20	12	10
G 15	20	14	10
G 16	20	15	10
G 17	20	11	11
G 18	20	13	12
G 19	20	15	11
G 20	20	13	12

VPC = Violation of Primary Constraint

5.5.2 Test 2: population size = 100, generation number = 40

The chart below shows the performance of the MSTTP-GA on test 2 (see figure 5.6). The best chromosome implemented 55% of the soft constraints with 40 generations (see table 5.3).

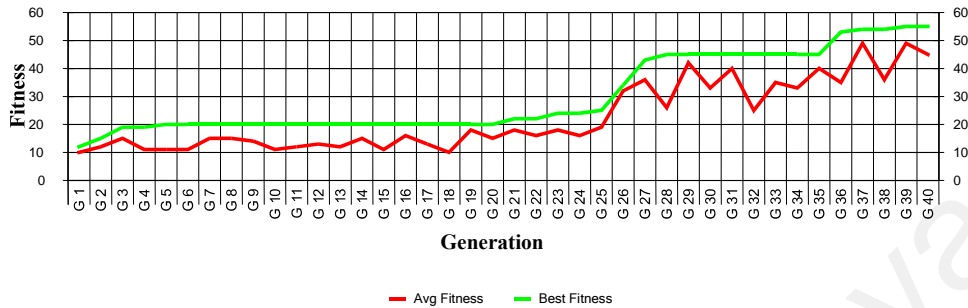


Figure 5.6: Case Study Test 2 Result

Table 5.3: Case Study Test 2 Result

Generation	Best	Avg	VPC	Generation	Best	Avg	
G 1	12	10	9	G2 1	22	18	12
G 2	15	12	10	G22	22	16	11
G 3	19	15	11	G 23	24	18	12
G 4	19	11	10	G 24	24	16	12
G 5	20	11	9	G 25	25	19	13
G 6	20	11	9	G 26	34	32	10
G 7	20	15	10	G 27	43	36	15
G 8	20	15	10	G 28	45	26	15
G 9	20	14	11	G 29	45	42	16
G 10	20	11	11	G 30	45	33	16
G 11	20	12	11	G 31	45	40	13
G 12	20	13	9	G 32	45	25	15
G 13	20	12	10	G 33	45	35	16
G 14	20	15	10	G 34	45	33	16
G 15	20	11	11	G 35	45	40	17
G 16	20	16	10	G 36	53	35	17
G 17	20	13	12	G 37	54	49	17
G 18	20	10	12	G 38	54	36	21
G 19	20	18	12	G 39	55	49	20
G 20	20	15	12	G 40	55	45	25

5.5.3 Test 3: population size = 100, generation number = 60

The chart below shows the performance of the MSTTP-GA on test 3 (see figure 5.7). The best chromosome implemented 65% of the soft constraints with 60 generations (see table 5.4).

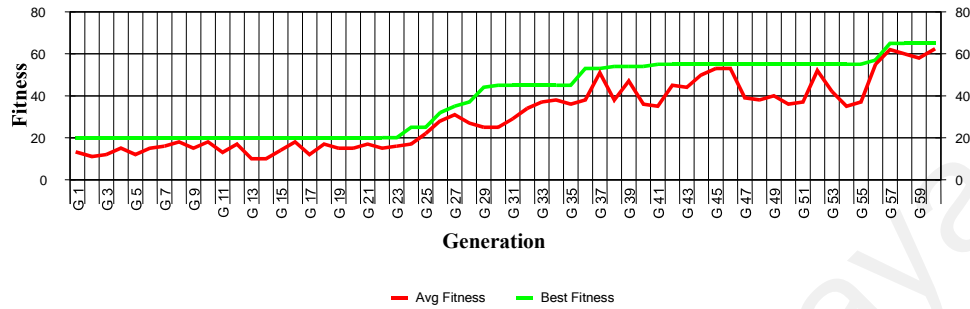


Figure 5.7: Case Study Test 3 Result

Table 5.4: Case Study Test 3 Result

Generation	Best	Avg	VPC	Generation	Best	Avg	VPC
G 1	20	13	13	G31	45	29	18
G 2	20	11	9	G32	45	34	16
G 3	20	12	12	G33	45	37	16
G 4	20	15	11	G 34	45	38	18
G 5	20	12	14	G 35	45	36	18
G 6	20	15	9	G 36	53	38	18
G 7	20	16	10	G 37	53	51	18
G 8	20	18	10	G38	54	38	18
G 9	20	15	11	G 39	54	47	16
G 10	20	18	11	G 40	54	36	18
G 11	20	13	11	G 41	55	35	16
G 12	20	17	8	G 42	55	45	18
G 13	20	10	11	G 43	55	44	16
G 14	20	10	10	G 44	55	50	17
G 15	20	14	12	G 45	55	53	17
G 16	20	18	12	G 46	55	53	18
G 17	20	12	11	G 47	55	39	21
G 18	20	17	13	G 48	55	38	21
G 19	20	15	11	G 49	55	40	21
G 20	20	15	12	G 50	55	36	21
G 21	20	17	10	G 51	55	37	21
G 22	20	15	10	G 52	55	52	21
G 23	20	16	12	G 53	55	42	21
G 24	25	17	14	G 54	55	35	21
G 25	25	22	14	G 55	55	37	21
G 26	32	28	14	G 56	57	55	21
G 27	35	31	16	G 57	65	62	21
G 28	37	27	16	G 58	65	60	21
G 29	44	25	16	G 59	65	58	21
G 30	45	25	18	G 60	65	62	21

5.5.4 Test 4: population size = 100, generation number = 100

The chart below shows the performance of the MSTTP-GA on test 4 (see figure 5.8). The best chromosome implemented 85% of the soft constraints with 100 generations (see table 5.5).

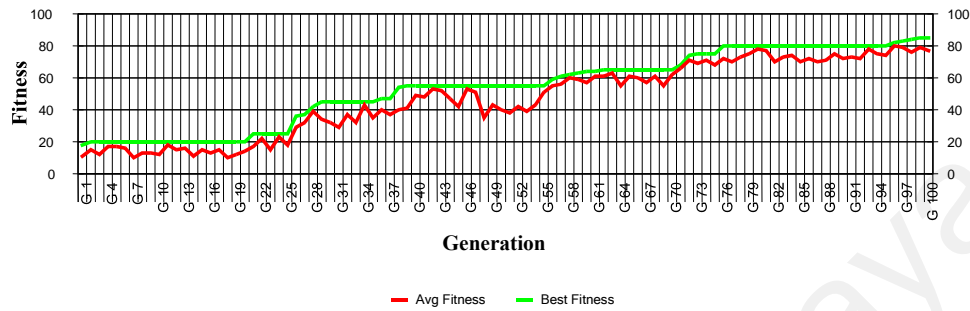


Figure 5.8: Case Study Test 4 Result

Table 5.5: Case Study Test 4 Result

Generation	Best	Avg	VCP	Generation	Best	Avg	VCP	Generation	Best	Avg	VCP
G 1	18	11	10	G31	45	29	17	G61	64	61	25
G 2	20	15	10	G32	45	37	15	G62	65	61	25
G 3	20	12	12	G33	45	32	16	G63	65	63	24
G 4	20	17	10	G 34	45	43	18	G 64	65	55	30
G 5	20	17	11	G 35	45	35	17	G 65	65	61	20
G 6	20	16	9	G 36	47	40	18	G 66	65	60	21
G 7	20	10	11	G 37	47	37	18	G 67	65	57	22
G 8	20	13	10	G38	54	40	16	G68	65	61	21
G 9	20	13	11	G 39	55	41	16	G 69	65	55	21
G 10	20	12	12	G 40	55	49	14	G 70	65	62	21
G 11	20	18	11	G 41	55	48	16	G 71	68	66	25
G 12	20	15	9	G 42	55	53	18	G 72	74	71	25
G 13	20	16	10	G 43	55	52	13	G 73	75	69	27
G 14	20	11	11	G 44	55	47	17	G 74	75	71	30
G 15	20	15	12	G 45	55	42	17	G 75	75	68	30
G 16	20	13	12	G 46	55	53	18	G 76	80	72	30
G 17	20	15	12	G 47	55	51	19	G 77	80	70	35
G 18	20	10	11	G 48	55	35	21	G 78	80	73	38
G 19	20	12	12	G 49	55	43	19	G 79	80	75	38
G 20	20	14	12	G 50	55	40	21	G 80	80	78	38
G 21	25	17	11	G 51	55	38	21	G 81	80	77	42
G 22	25	22	10	G 52	55	42	21	G 82	80	70	40
G 23	25	15	12	G 53	55	39	21	G 83	80	73	40
G 24	25	23	14	G 54	55	43	20	G 84	80	74	42
G 25	25	18	13	G 55	55	51	20	G 85	80	70	42
G 26	36	29	14	G 56	59	55	20	G 86	80	72	42
G 27	37	32	16	G 57	61	56	21	G 87	80	70	41
G 28	42	39	15	G 58	62	60	21	G 88	80	71	41
G 29	45	34	16	G 59	63	59	20	G 89	80	75	40
G 30	45	32	18	G 60	64	57	21	G 90	80	72	40

Table 5.5: Case Study Test 4 Result (Continued)

Generation	Best	Avg	VCP
G91	80	73	41
G92	80	72	41
G93	80	78	41
G 94	80	75	41
G 95	80	74	41
G 96	82	80	40
G 97	83	79	42
G98	84	76	43
G 99	85	79	42
G 100	85	77	43

To review the master timetable of the 4 best chromosomes produced by test1, test2, test3, and test4 (see appendix B)

Table 5.6: shows the Simple genetic algorithm results on test1, test2, test3, and test4, population size=100 and using four different generations. The last three columns represent worst, average, and best chromosomes. The values 12%, 20% ... etc are the fitness level of the chromosomes. For example, the best chromosome fitness in test 1 is 28% means that the chromosome implements the soft constraints by 28% (see table 5.6). Note that all hard constraints are mandatory met.

Table 5.6: Simple Genetic Algorithm Results

Test	Generation	Start time	End time	Avg Chromosome	Best Chromosome
1	20	3:08:03	3:11:17	12%	20%
2	40	3:30:00	3:36:30	30%	55%
3	60	4:09:25	4:19:09	35%	65%
4	100	5:23:19	5:39:36	45%	85%

5.6 Discussion

Basically, there are three factors which affect the system Initial population, number of generations and the quality of data provided by the user. They are discussed below in detail.

5.6.1 Population Size

Different population sizes have its advantage and disadvantage. The advantage of large population includes, good chromosomes will have chances to reproduce, but converges much slower. For example, if the population size is 300, the best chromosome will have many chances to mate with other chromosomes, and it will take many generations before it converges. While, if the population consists of only 10 chromosomes, it will converge very fast. Since after few generations the 10 chromosomes will comprise quite the same genes and at the same time the best chromosome will have one chance only to reproduce, the time taken to execute the genetic algorithm of this large population size will be longer than the time taken to execute the same GA of a small population size and yet yielding almost the same results. In general, large population gives better solution. But at the same time, as the size of the population increases, the execution time increases, and the difference diminishes.

5.6.2 Generation Numbers

Generation numbers have the same advantages and disadvantage on the GA as the population size. The only difference between them is that if the size of the population is small it allows a good chromosome to have a chance to mate many times compared to with a large number of generations would give good chance to mate the good chromosome with small size of population.

5.6.3 Data Quality

Obtaining good chromosome fitness has a closed relationship with the data entered by the user as discussed in section 4.2 (chapter 4). Number of lab rooms, number of teachers teaching particular subject or particular subject taught by number of teachers, and number of

periods of different subject, all are affecting the final result of the school timetable negatively or positively.

5.7 research contribution

The main contributions of this project can be summarized as follows:

- Soft constraints

As a result of survey we come out with number of soft constraints which we reform them in mathematical formulas as listed below :-

1. Science subjects must be scheduled early in the timetable. $(c_n) = (\sum_{i=1}^k V_c) \times w_c$ (See section 3.2.3.2.2)

2. Some lessons may only be scheduled to particular periods. $WT = \sum_{day1}^5 CSV \times W$ (See section 3.2.3.2.2)

3. Teachers prefer to have more lessons on some days, in order to have a day without lessons (free day). $(c_n) = (\sum_{i=1}^k V_c) \times w_c$ (See section 3.2.3.2.2)

4. Lessons of the same subject for a class must be distributed uniformly over the week.

$$DF = \sum_{c_1}^{c_n} c \times \sum_{t_1}^{t_n} tv \times W \text{ (See section 3.2.3.2.2)}$$

For more details see (chapter 3 section 3.2.3.2.2)

- Developed Tool (MSTP-GA)

The second contribution of this project is the school timetable which we name MSTP-GA. The tool is designed based on genetic algorithms.

5.8 MSTTP-GA Evaluation

One way to assure that validation of a software system, is to evaluate its compliance with its requirements. This section presents MSTTP-GA evaluation process. It describes the pilot study conducted to evaluate its features and requirements.

In addition to the heavy white and black box testing accompanied MSTTP-GA development life cycle, It has been evaluated by group of eleven of Malaysian secondary schools (see: Table 5.1). the pilot study conducted is explained below.

5.8.1 Pilot study

Our pilot study includes a number of extensive knowledge teachers about school timetabling. The whole group have practice school timetable before. The study includes qualitative measurements include the assessment of MSTTP-GA by its users collected using a questionnaire (see: Appendix E)

5.8.1.1 Participants and experiment material

The study involves eleven of Malaysian secondary schools. Two to three teachers in charge of timetabling evaluated the tool in each school. All participants have extensive knowledge about school timetabling.

5.8.1.2 Environment

Most of participants used the computer at school. The computers specification ranged from Pentium II 266 MHz with 128 Megabyte RAM to Pentium III 480 MHz with 256 Megabyte RAM. The operating system used are windows 98, windows Me, windows 2000.

5.8.1.3 Methodology

It has been shown that all of the users are strongly satisfy with the final result of tool but they are get problem about understanding of GAs operators. For such reason, each group was briefed about Genetic algorithms operators before using MSTTP-GA. Number of questionnaires passed to each school depends on the number of teacher who participant to evaluate the tool.

5.8.1.4 Qualitative measurements

Participants were asked to fill in the questionnaire. The questionnaire is organized to cover the whole features implemented in MSTTP-GA. The answer received (see figure 5.9) are discussed below:

Figure 5.9 shows the summary of questionnaire and its results. The mean generally scored above the average in almost all the questions.

Table 5.7 A and figure 5.9 A shows that 73.7% of the users rate the system as excellent tool for timetabling.

Table 5.7: Q1

How would you rate the system?

		Frequency	Percent
Valid	Good	5	26.3
	Excellent	14	73.7
	Total	19	100.0

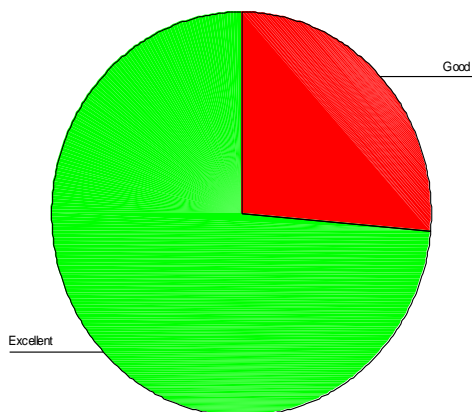


Figure 5.9 : Q1

Table 5.8: Q2

The system is easy to learn and use.

		Frequency	Percent
Valid	Strongly agree	6	31.6
	Agree	13	68.4
	Total	19	100.0

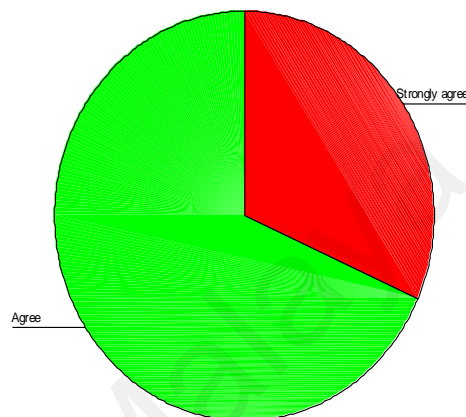


Figure 5.10 : Q2

Table 5.9: Q3

System interface is friendly.

		Frequency	Percent
Valid	Agree	9	47.4
	Strongly agree	10	52.6
	Total	19	100.0

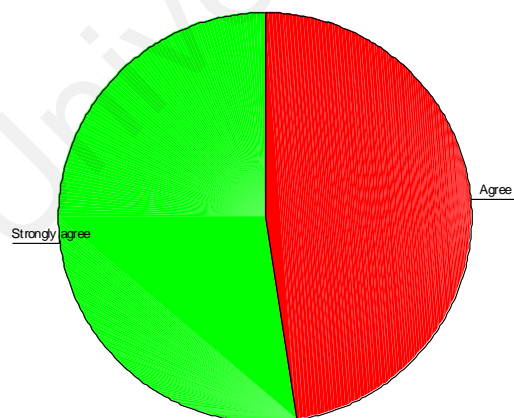


Figure 5.11 : Q3

Table 5.10: Q4

The system is helpful in assigning teachers to classes depending on specialization.

		Frequency	Percent
Valid	Agree	5	26.3
	Strongly agree	14	73.7
	Total	19	100.0

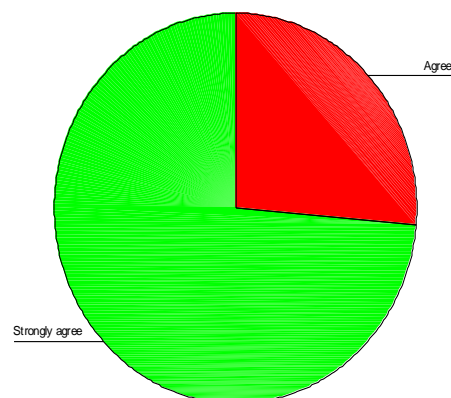


Figure 5.12 : Q4

Table 5.11: Q5

System input/output helpful and easy

	Frequency	Percent
Valid Strongly agree	4	21.1
Agree	15	78.9
Total	19	100.0

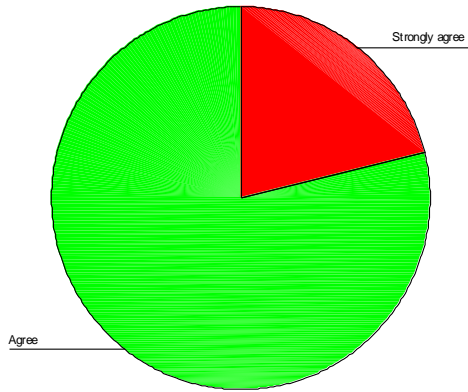


Figure 5.13 : Q5

Table 5.12: Q6

With this system, less time is spent on input, scheduling, and output data

	Frequency	Percent
Valid Strongly agree	6	31.6
Agree	13	68.4
Total	19	100.0

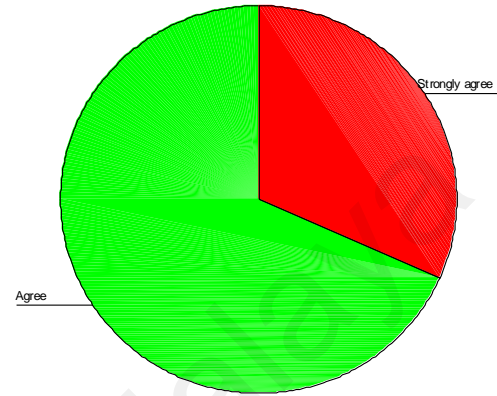


Figure 5.14 : Q6

The answers convey that almost all the participants agree that it is over average and none of them stated it is fairly or extremely unusable.

Chapter 6: Conclusion and future work

6.1 Introduction

This chapter summarizes the work of the project, present the conclusions and provide possible future work. Future work includes several possible extensions to the current work.

6.2 Conclusion

A school timetabling system using GA is developed in this project. The success of this project is to achieve the main objective that is defined in section 1.6 chapter one.

This project has investigated several issues related to the Genetic Algorithms and methodologies for Constraints Satisfaction Problems particularly oriented to solving the weekly school timetabling problem.

The developed system can initialise population of school timetable solutions that are feasible (satisfies to hard constraints) all the time.

The feasible solutions produced by system are passed to simple GA and evaluate each solution individually to make them satisfy to the soft constraints.

The system user interface is very friendly and easy to use. System hides the complex components from users and the only things that user has to provide to system are size of population, generation and some GA operator like type of crossover and selection methods.

The result of the system is depends on size of population, generation iteration times, and quality of data.

6.3 Future Work

6.3.1 GA's suggestions

In this project, basic Genetic Algorithms used to develop school timetable tool for Malaysian secondary schools. GA's operators used were one and two point crossover, Roulette Wheel Parent Selection, Rank Based Selection, normal mutation. For this part different GA's techniques and operators can be used as multipoint crossover, variable and direct mutation, Tournament Selection, Uniform Random Selection.

6.3.2 Problem representation

In GA's, the method of chromosome representation and its parameters considered as base to built robust GA's application. In this problem chromosome represented as two dimensional array each row represent one class, each eight column represent one day, each day divided to eight periods, each period considered as gene which consist of two element (teacher/subject). This method can be used again with applying small change to the chromosome representation. 2D array whereas rows represent teachers and (class/subject) represent the gene.

List of Appendices

Appendix A: Reference

Appendix B: Experimental Materials

APPENDIX C: Questionnaire Of System

Appendix D: Result of Survey

Appendix E: Questionnaire of MSTTP-GA Tool

University of Malaya

Reference

A. Mehrotra and M. Trick (1996). A column generation approach for graph coloring. *INFORMS Journal On Computing*. vol. 8, no. 4, pp. 344--354, 1996

A. Schaerf (1996). Tabu search techniques for large high-school timetabling problems. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 363 -- 368, Portland, Oregon, August, 1996.

Abramson and J. Abela,(1992) A Parallel Genetic Algorithm for Solving the School Timetabling Problem. Appeared in 15 Australian Computer Science Conference (ACSC-15), vol. 14, p. 1--11, 1992., Hobart, Feb 1992.

Abramson, D.A., Dang, H. and Krisnamoorthy, M. (1999). Simulated Annealing Cooling Schedules for the School Timetabling Problem, *Asia-Pacific Journal of Operational Research*, 16 (1999) 1-22

Alberto Colomi, Marco Dorigo, and Vittorio Maniezzo(1993): A genetic algorithm to solve the timetable problem Technical Report No. 90-060, Politecnico di Milano, Italy 1993, 24p

Bambrick (1997), *Lecture Timetabling Using Genetic Algorithms*, thesis, Department of Electrical and Computer Engineering, The University of Queensland.

Burke E. K, D. G. Elliman, and R. F. Weare, (1995) "A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems," 6th International Conference on Genetic Algorithms (Pittsburgh, USA, 15- 19 July 1995). pages 605-610, San Francisco. Morgan Kaufmann.

Burke E. K., Elliman D. G. and Weare R. F. (1994), A University Timetabling System Based on Graph Colouring and Constraint Manipulation, *Journal of Research on Computing in Education*. . Vol. 27. issue 1, pp1-18.

Burke E. K., J. A. Clark, and A. J. Smith (1997). Four methods for maintenance scheduling. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 264--269. Springer, 1997.

Back, T.(Ed.)(1997).*Proceedings of the Seventh International Conference on Genetic Algorithms* .San Francisco, CA: Morgan Kaufmann.

Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., & Smith, R. E. (Eds.) (1999). *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*. San Francisco, CA: Morgan Kaufmann.

C. Blum, M. Dorigo, S. Correia, O. Rossi-Doria, B. Paechter, and M. Snoek (2002). A GA evolving instructions for a timetable builder. In *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling (PATAT 2002)* (to appear), 2002.

Chambers (1995), *Practical Handbook of Genetic Algorithms*, Chapter 8, *Evolving Timetables. Application Vol1*, p 219-222

Chan Yam Ling, (1994). *A Genetic Algorithms Shell for Iterative Timetabling*, thesis. Department of computer science RMIT.

D. Beasley, D. R. Bull, and R. R. Martin (1993a). "An Overview of Genetic Algorithms: Part I, Fundamentals," *University Computing*, vol.15, no.2, pp. 58—69.

David E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Publishing Co., Inc. 1989

D. Corne E. Collingwood, and P. Ross (1996). *A Guide to GATT*. Department of Artificial Intelligence, University of Edinburgh,.

David A Coley (2001). *An Introduction to Genetic Algorithms for Scientists and Engineers*. University of Exeter. World Scientific Publishing Co. Pet. Ltd

David Beasley, David R. Bull, and Ralph R. Martin (1993b). An overview of genetic algorithms: Part 2, research topics. *University Computing*, Vol.15, No.4, pp.170--181.

David E Goldberg *Genetic Algorithms in Search Optimization and Machine Learning* Addison Wesley, 1989.

Davis (1991): *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold. p4

Edmund Kieran Burke and Sanja Petrovic, (2002). *Recent Research Directions in Automated Timetabling*. Accepted for publication in

European Journal of Operational Research – EJOR, Vol 140/2, 2002,p 266-280.

Eiben, A. E., Back, T., Schoenauer, M., & Schwefel, H.-P.(Eds.)(1998).Parallel Problem Solving from Nature,PPSN V .Berlin:Springer-Verlag.

E.K. Burke, D.B. Elliman & R. Weare, 94), A University Timetabling System based on GraphColouring and Constraint Manipulation, University of Nottingham., 1994

Fogel, D.B., Schwefel, H. P.,Back, T., & Yao, X.(Eds.)(1998).Proceedings of 1998 IEEE International Conference on Evolutionary Computation .Piscataway, NJ: IEEE Service Center.

Falkenauer Emanuel (1999). Genetic Algorithms and grouping Problems. Department of Applied Mechanics, University Libre de Bruxelles. John Wiley & Sons

Fang, H.L.,(1994).Genetic Algorithms in Timetabling and scheduling. Ph.D thesis, Department of AI, University of Edinburgh

G. R. Filho and L. A. N. Lorena (2001). A Constructive Evolutionary Approach to School Timetabling. In Boers et al. E. J. W. Boers et al., editor. Springer Lecture Notes in Computer Science, volume 2037. Springer-Verlag, April 2001. , pages 130-139.

G. R. Harik and F. G. Lobo (1999). A parameterless genetic algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 258--265, 1999. 5. A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 3(2):124--141, July 1999.

Glover F. and Laguna M. (1993). Chpter 3: tabu serch. Modern Heuristic Techniques for Cominatorial Problems.

L. Davis, Handbook of Genetic Algorithms Van Nostrtrand Reinhold New York, 1991.

John Holland *Adaptation in Natural and Artificial Systems* The University of Michigan Press Ann Arbor , 1975.

Kiang, (1998). Application of GA in optimizing the examination timetable problem university Malaysia Sarawak. Thesis. Faculty of Information Technology. University of Malaysia Sarawak.

Koza, J.R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D.B., Garzon M.H., Goldberg, D.E., Iba, H., & Riolo, R.L. (Eds.) (1998). *Genetic Programming 98*. San Francisco: Morgan Kaufmann Publishers.

Melicio Fernando, João Paulo Caldeira, and Agostinho C. Rosa (1999). Solving the Timetabling Problem with Simulated Annealing. *ICEIS 1999*: 272-279

Michalewicz, (1999). *Genetic algorithms + Data Structures = Evolution Program*. Department of Computer Science, University of North Carolina. Third, Revised and Extended Edition. Springer

Mukherjee (2001). *Timetabling using Cellular Genetic Algorithms with Adaptive Mutation Operators*. Thesis. Department of Computer Science University of York

M.A. Saleh Elmohamed and Geoffrey Fox, 98). *A Comparison of Annealing Techniques for Academic Course Scheduling*, Department of Computer Science University of Adelaide, Adelaide 4 April 1998

R A Fisher, *The Genetical Theory of Natural Selection* Dover Books, 1958

Reis, L. P. and Oliveira, E (2000). "A Language for Specifying Complete Timetabling Problems. *Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling*, pp.456-471, Konstanz, Germany, August 16-18 2000

Ross Peter, Dave Corne, and Hsiao-Lan Fang (1994). Successful lecture timetabling with evolutionary algorithms. In *Proceedings of the 11th European conference on Artificial Intelligence, Amsterdam Workshop on Evolutionary Algorithms*, SpringerVerlag 1994.

S. Elmohamed, P. Coddington, and G. Fox (1997). A comparison of annealing techniques for academic course scheduling. In E. Burke and M. Carter, editors, *Proceedings of the 2nd International Conference on the Practice and Theory of Automated Timetabling PATAT '97*, LNCS 1408, pages 92--112. Springer-Verlag, 1997.

Sandikci Burhaneddin (2000).Genetic Algorithms, Bilkent University
Department of Industrial Engineering 06533, Ankara-Turkey

Schaerf (1995). A Survey of Automated Timetabling. Centrum voor
Wiskunde en Informatica (CWI) report CS-R9567, Amsterdam, The
Netherlands. A revised version appeared in Artificial Intelligence Review
13(2), 87-127.

Schaerf Andrea and Gaspero Luca (2001). Local Search Techniques for
Educational Timetabling Problems , Proceeding of the 6th International
Symposium on Operational Research in Slovenia (SOR-'01), Preddvor,
Slovenia, (pp.13-23).

Voráč, J, I. Vondrák, K. Vlček (2001). School Timetable Generating using
Genetic Algorithm, VSB – Technical University of Ostrava, Czech
Republic

Willemen Robertus J., (2002). School timetable construction: Algorithms
and complexity, (Eindhoven University of Technology) Dissertatie
Eindhoven.