ASSIGNATION OF PSM EVALUATOR USING GENETIC ALGORITHM

YAP SUET LEE

UNIVERSITI MALAYSIA PAHANG

ASSIGNATION OF PSM EVALUATOR USING GENETIC ALGORITHM

YAP SUET LEE

A report submitted in partial fulfillment

of the requirements for the award of the degree of

Bachelor of Computer Sciences (Graphic & Multimedia Technology)

Faculty of Computer System & Software Engineering

Universiti Malaysia Pahang

JUN 2012

SUPERVISOR'S DECLARATION

"I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of the degree of Bachelor of Computer Sciences (Graphic & Multimedia Technology)"

Signature	:	
Name of Supervisor	:	ZALILI BINTI MUSA
Position	:	LECTURER
Date	:	11 JUN 2012

STUDENT'S DECLARATION

I declare that this thesis entitled "Assignation of PSM Evaluator using Genetic Algorithm" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : YAP SUET LEE

ID Number : CD09026

Date : 11 JUN 2012

ACKNOWLEDGEMET

I am grateful and would like to express my sincere gratitude to my supervisor Miss Zalili Binti Musa for her great ideas and invaluable guidance in making this project possible. She also gives me continuous encouragement and constant support while doing the project. I appreciate her hard work to guide me from the first day I approach to her to have her as my supervisor to these concluding moments. I am truly grateful for her progressive vision about my training in writing good documentation and she often impressed me with creative idea in designing the system flow. I often make naïve mistakes but she able to tolerance with me and lead me back to the correct way.

My sincere thanks go to the lecturers of my faculty who helped me directly or indirectly to complete this project. I also would like to express a very special thanks to the evaluators of my project for their suggestions and opinions. I also sincerely thanks to all my friends in the university who helped me in many ways and their excellent co-operation, inspiration and supports.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals and studies. They consistently encouraged me to carry on my studies and make achievements in life.

ABSTRACT

The purpose of this paper is to present a design of development for Assignation of PSM Evaluator using Genetic Algorithm (APEGA) system. This is an application system that is used to assist the Faculty of Computer System and Software Engineering (FSKKP) of University Malaysia Pahang (UMP) in matching the optimum evaluators for the students in PSM presentation carnival. In the methodology part, a development model which involves with client participation is designed in order to use in the development of this project. The target user of the system is PSM coordinator who is responsible in assigning the PSM evaluator. Assignation of PSM Evaluator using Genetic Algorithm (APEGA) is expected to be able in developing a well-distributed matching and overcoming the relevant constraints in an intelligent way. Therefore, it is tend to reduced human energy compare to the current manually assign the PSM evaluator. The main purpose of this application system is to solve the optimization problem that occurred during the assignation of PSM evaluator. For this system, it is able to assign the evaluator to evaluate on the students' project which related with his or her expertise field.

ABSTRAK

Tujuan kajian ini adalah untuk membentangkan reka bentuk pembangunan Sistem Pengagihan Penilai PSM menggunakan Algoritma Genetik (APEGA). Sistem aplikasi ini adalah untuk membantu Fakulti Sistem Komputer dan Kejuruteraan Perisisan (FSKKP) memadankan penilai yang optimum bagi pihak pelajar untuk Karnival Pembentangan PSM. Dalam bahagian metodologi, suatu model yang terkandung dengan penglibatan klient telah direka untuk digunakan dalam pembanguan projek ini. Pegguna sasaran sistem ini adalah penyelaras PSM yang bertanggungjawab dalam pengagihan penilai PSM. Sistem Pengagihan Penilai PSM menggunakan Algoritma Genetik (APEGA) dijangkakan dapat menghasilkan agihan yang memadan dengan bagus dan mengatasi kekangan yang berkaitan dengan secara pintar. Oleh itu, ia cenderung untuk mengurangkan kegunaan tenaga manusia berbanding dengan pengagihan penilai PSM secara manual. Tujuan utama system aplikasi ini adalah untuk menyelesaikan masalah pengoptimuman yang berlaku semasa pengagihan penilai PSM. Sistem ini dapat mengagihkan penilai-penilai untuk menilaikan projek pelajar yang berkaitan dengan kepakaran mereka.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

APEGA	Assignation of PSM Evaluator using Genetic Algorithm
PSM	Projek Saujana Muda
GA	Genetic Algorithm

CHAPTER 1

INTRODUCTION

1.1 Introduction

Assigning evaluator is same like scheduling process which is common but complex. It is quite hard to create an optimum schedule because there here are a lot of criteria and constraints need to be considered when scheduling a timetable. Therefore, there is a need to have a scheduling optimization by using some techniques. As a preparation in front on era technology information like today, implementation of technology and artificial intelligent in management is playing an important role in order to obtain an optimum way in scheduling a timetable.

Some of the artificial intelligent methods are suitable to use in optimize schedule such as Expert System, Neural Network and Genetic Algorithm. In this project, Genetic Algorithm (GA) technique will be used to solve the optimization problem. GA is the most suitable technique use to optimizing on both small and large schedules for schedule optimization. This is because GA was often used to conquer nondeterministic polynomial (NP) problem, thus GA has better quality of the optimization.

Therefore, the Assignation of PSM Evaluator system is developed to assist the Faculty of Computer System and Software Engineering (FSKKP) in matching the optimum evaluators for the students for PSM presentation. In achieving the goal, Genetic Algorithm will be implemented into Assignation of PSM Evaluator application, thus the system can optimize the assignation of PSM evaluator for the student based on their project type and evaluator's expertise.

Assignation of PSM Evaluator system by using technique of Genetic Algorithm will be paperless and tend to reduced human energy. Indeed, the problem of clashing between student and other factors can be minimized with the aid of the system, thus an optimum PSM evaluator assignation's list can be produce instead of the current existing application which manually assign the PSM evaluator. Basically, the system will be implemented with the required rules and constraint that need to be compiled to developing a well-distributed matching by overcoming the relevant constraints in an intelligent way.

1.2 Problem Statement

Matching the optimum evaluators for the students is not an easy task because it has its complex constraints optimization problem. Thus, there are a lot of problems and clashing occurs when assigning the evaluators for PSM presentation for Faculty of Computer System and Software Engineering (FSKKP). The previously assignation of PSM evaluator for PSM presentation that done by PSM coordinator are lack of optimum and contain a lot of clashing such as available of evaluators. This is because human is inability to think out a more optimum schedule without using certain technique. Besides that, due to the number of evaluator is limited, so one evaluator has to evaluate quite a number of student's project. Thus, it is difficult to arrange the evaluator and the project that they are going to evaluate has relevant to their field or expertise.

There are a lot of constraints to be concern when generating PSM presentation's schedule. The constraints which have to consider are such as the number of students that who are going to present their project and the number of lecturers that who will be the evaluator for PSM presentation. Besides that, the project title which the students going to present and lecturer's expertise are the major constraints that have to be concern. This is because there are always happened on the previous assignation that some of the lecturers had been arranged to evaluate on the student's projects which were not under their own field or expertise. The limited of place or laboratory are also one of the

constraints that in scheduling. Thus, it will be the problem that in order to assign the optimal evaluators for the student.

In additional, there are few constraints are changeable and different in every year such as the student's project title, lecturers that some had leaving or some are new comer that have to be reconsider every year for matching the optimum evaluators for the students. Therefore, it is take time to update the new data or rearrange it by manually. There are a lot of difficulties had been found in assigning the PSM evaluator by human energy compare with a computerized system that had implemented with Artificial Intelligence technique .

1.3 Objectives

The objectives of this project are:

i. To study Genetic Algorithm technique.
ii. To develop a prototype in matching PSM evaluator using Genetic Algorithm for assigning the optimum evaluators for the student in PSM presentation.

1.4 Scope

The scopes of this project are:

(i) The users of the application are PSM coordinator of Faculty Computer System and Software Engineering of University Malaysia Pahang (UMP) who responsible in assign the PSM evaluator.
(ii) Total of 12 lecturers and approximate 30 students will be used in this project.
(iii) All the project's titles for PSM student are distinguish in four main categories which are Computer Science (Management Information

System), Computer System and Networking, Software Engineering and Graphic and Multimedia.

(iv) The application will be develop by using Microsoft Visual Studio 2010 and Microsoft SQL Server 2008 is used to store the system database.

(v) The Assignation of PSM Evaluator using Genetic Algorithm (APEGA) application is standalone system.

1.5 Thesis Organization

This thesis consists of six chapters and each chapter is to discuss the different issues in the project. Below that is the summary of the content for each chapter.

i. Chapter 1 – Introduction
   - This chapter provides background information about the project which includes problem statement, objectives and scope.

ii. Chapter 2 – Literature Review
   - Some literature and research which related to this project will be review and discuss in this chapter.
   -

iii. Chapter 3 – Methodology
   - Data analysis, method and the procedure of this project development will be discussed.

iv. Chapter 4 – Implementation
   - The implementation of the system using Genetic Algorithm will be explained in this chapter.

v. Chapter 5 – Result and Discussion
   - This chapter will present the testing result of the system and result on the discussion.

vi. Chapter 6 – Conclusion

- A complete summary of the project will be present in this chapter.

CHAPTER 2

LITERATURE REVIEW

This chapter is a review of the literature that discusses to identify studies relevant to the topic. The purpose of this chapter is to know more about how the implementation of Genetic Algorithm in application. In this chapter, the background of Genetic Algorithm technique will be discussed followed by and existing application that using Genetic Algorithm technique.

Comment [L1]: tense

2.1 Scheduling

Scheduling is a constraint satisfaction activity. S. F. Smith had proposed that there are three most important types of scheduling constraints which are time constraints, resources constraints, and casual constraints were caused by internal and external factors [1]. Scheduling involves making decision regarding to the allocation of available capacity or resources such as equipment, labour and space to tasks, activities over time. It is seeks to achieve several objectives that which are high efficiency and low inventories. Scheduling is a very important activity. An effective scheduling in manufacturing and office environment can increase productivity and decrease cost by significant factor [2]. Thus, scheduling system has been widely used across many field of expertise to schedule for optimize usage of resources such as in manufacturing sites, schedule timetabling for school and workplace.

In the earlier days, scheduling was a time-consuming task which has to take a lot time to typing up the information and schedule out a timetable without any clashing of constraints. With the advent of high computer technologies and flatter hierarchies, scheduling can be done with the aid of few intelligence techniques to achieve

performance goal in minimizing execution time and communication delay while maximizing the resources utilizations to produce an optimum schedule.

2.1.1 Scheduling Problem

A scheduling problem can be defined as the problem of assigning a number of events into a limited number of periods [3]. There are various problems of scheduling and sequencing which has been addressed since 1950's by researchers in computer science, operation research and discrete mathematics [4]. When applying a schedule in a dynamic environment, it is hard to fulfill the requirements the first time round based on some scheduling problems and its constraints. Scheduling problem is uncertain and changeable over the time. The environment of scheduling, scheduling act and the human resource affecting the scheduling are the three main factors that concern scheduling problem.

In order to solve the scheduling problem, one must develop a comprehensive scheduling system, which the system should be able to perform the following functions [5]:

- Absorb and accommodate unforeseen changes.
- Obtain visual information and knowledge of scheduling, and record all variables.
- Complete intelligent scheduling strategies.
- Produce real-time scheduling solution.
- Control the scheduling tasks in a dynamic environment.

For the scheduling problems which are subject to many constraints are usually divided into two categories: "hard" and "soft" [6]. Hard constraints are rigidly enforced while soft constraints are those are desirable but not absolutely crucial and it is usually hard to satisfy all the soft constraints in real-time situation [3].

There are a set of large number of all possible solutions appearing due to its large number of events which to be scheduled and a wide variety of constraints that imposed on scheduling. Thus, it is extremely difficult to generate a timetable and its manual solution is requiring much more effort.

2.2 Genetic Algorithm

Beasley et al. had stated that Genetic Algorithms are adaptive method which used to solve search and optimization problem [9]. Genetic Algorithms are often know as an optimizer tool for a widely areas of research including the field of system control and control design. Extensive research has been done and shows the exploitation of the Genetic Algorithms which have robust properties and demonstrating their capabilities across a broad range of problems [10]

In the early 1970s, the concept of Genetic Algorithm (GA) is introduced by John Holland who is a professor of psychology at University of Michigan [11]. Genetic Algorithm is a family of computational models inspired by Darwin's Theory of Evolution [12]. Genetic Algorithm is a stochastic search technique that guides a population of solution using the principles of evolution and natural genetic [13]. Charles Darwin had claimed that the natural populations evolve over successive generations based on the principles of natural selection example, survival of the fittest [12]. The operations of Genetic Algorithm such as reproduction, crossover and mutation are performed on the population and evaluated each individual's fitness [14]. A potential solution is encode to a specific problem on a simple chromosome-like data structure by Genetic Algorithm and the recombination operators is apply to these structures as to preserve the critical information [10].

2.2.1 Genetic Algorithm Procedures

In the initial stage of Genetic Algorithm, a population is generated randomly and comprises a group of chromosomes. The chromosomes are evaluated by calculated its

fitness values. From the population, a particular group of the chromosomes (parents) is selected to generate the offspring by genetic operations. The offspring's fitness is evaluated as like their parents and they will replace the chromosomes in the current population. All of these procedures of Genetic Algorithm are executed in a cycling process and it is repeated until reached a desired termination criterion [15]. Ursula Fissgus had stated in her research that the fitter chromosomes have higher probabilities of being selected [13]. After several generations, the best chromosome in the final population will become a highly evolved solution to the problem.

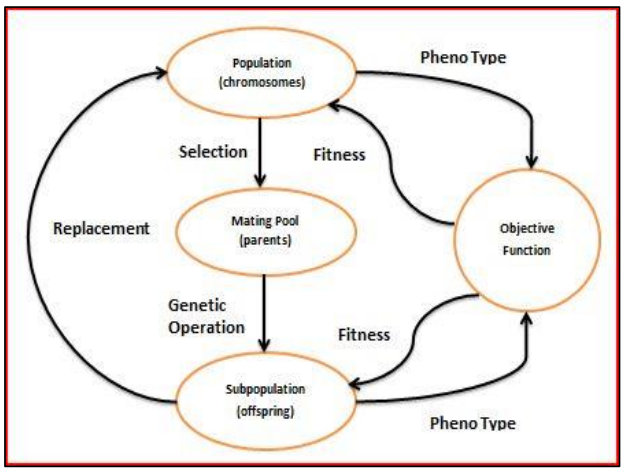


Figure 2.1: GA Cycle [15]

There are the outlines of the basic Genetic Algorithm that had discussed in the research paper by Obitko Marek [16]:

1. **[Start]** Generate random population of n chromosomes (suitable solution for the problem)
2. **[Fitness]** evaluate the fitness *f(x)* of each chromosomes x in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection**] Select two parents chromosomes from a population according to their fitness (the better fitness, the biger chance to be selected)

2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents
3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome)
4. **[Accepting]** Place new offspring in a new population

4. **[Replace]** Use new generated population for a further run of algorithm
5. **[Test]** If the end condition is satisfied, stop, and return the best solution in current population
6. **[Loop]** Go to step2

2.2.2 Genetic Algorithm Operations

The operation of Genetic Algorithm is start with a population of a random string representing decision variables [13]. A genetic operator is a process that used in Genetic Algorithms to maintain genetic diversity. Three main Genetic Algorithm operators which are reproduction, crossover and mutation operated the population to create a new population of points.

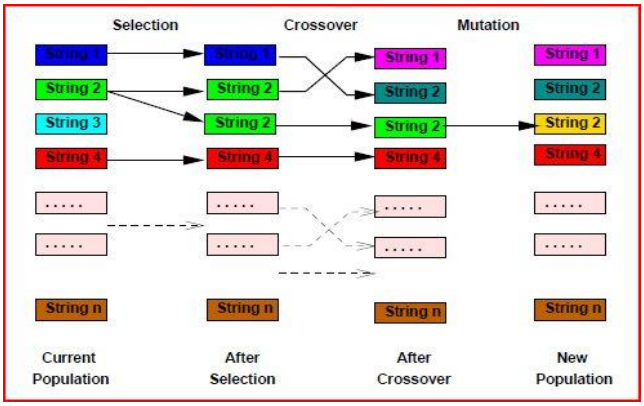


Figure 2.2: The Basic of Genetic Algorithm Operations. One generation is broken down into a selection phase and recombination phase. Strings are assigned into adjacent slots during selection [13].

2.2.2.1 Encoding

Genetic Algorithm starts with encoding the chromosome and code the candidate solution of an optimization algorithm. There are three commonly type of encoding techniques that are used in Genetic Algorithm field which are bit string encoding, permutation encoding and three encoding. Binary encoding is the most common type of encoding that are in use due to it can produce many possible chromosomes even with just a small number of alleles.

Chromosome A	10110010110010101110010 1
Chromosome B	11111110000011000001111 1

Figure 2.3: Example of Encoding for Two Chromosomes

2.2.2.2 Selection

Selection is an operation of Genetic Algorithm that randomly selected chromosomes of the population based on their fitness and forms a mating pool [18]. The higher fitness value chromosomes, the more likely it will be selected. Natural selection causes the chromosomes which encode successful structures more frequent to produce its copies. Tom V. Mathew had said that it is necessary to have selection operation of the chromosomes in the current population to maintain the generation of new population [10]. Roulette-Wheel selection is the common technique that use in Genetic Algorithm selection. In Roulette-Wheel selection, a chromosome is selected for the mating pool based on the probability proportional to its fitness.

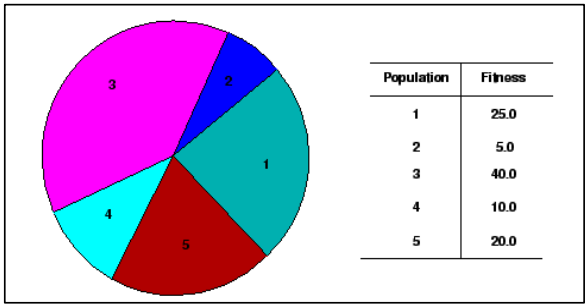


Figure 2.4: A Roulette-Wheel that Represents of Five Individual and Their Fitness Values [10]

2.2.2.3 Crossover

Crossover as a recombination operator that used to produce offspring that contain some parts of both parents' genetic material by combining subparts of two parent chromosomes [15]. In the crossover operator, the child is formed by a copy of first parent then the crossover position is randomly selected and the information of the second parent is introducing into the child chromosome. Ursula Fissgus discussed that different crossover operations are developed for different parts of the chromosome.

i. One Point Crossover

A crossover point on the parent chromosome is randomly selected. All information beyond that point within the two parent chromosome is swapped and interchanges to produce two new offspring.

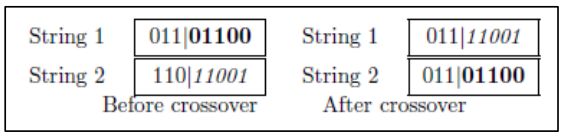


Figure 2.5.1: One Site Crossover Operation [10]

ii. Two Point Crossover

Two crossover points on the parent chromosome are randomly selected. The information in between the two points within two parent chromosome is swapped and interchanges to produce two new offspring.

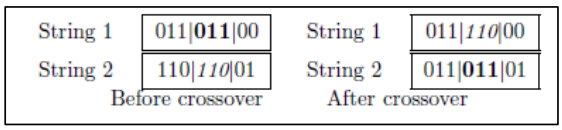


Figure 2.5.2: Two Site Crossover Operation [10]

2.2.2.4 Mutation

Mutation is a genetic operator that used to maintain genetic diversity from one generation to the next and avoid the population from stopping at any local optima [18]. The operation of mutation produces a new chromosome by randomly changing a randomly selected chromosome of the population. The position of the chromosome also randomly chooses for mutate. Ursula Fissgus claimed that the mutation operations not really ensure to achieve the change of the chromosome [13].

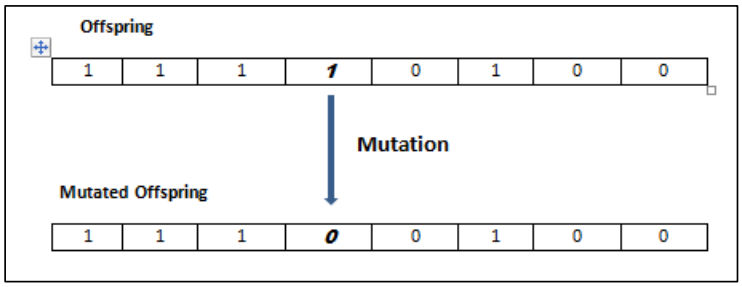


Figure 2.6: Mutation generation

2.3 Existing Genetic Algorithm Application

The objective of this section is to discuss about the research on the existing application which using Genetic Algorithm as the implementation technique. There are two system will be describe which are Genetic Algorithm Class Schedule and Faculty Timetabling using Genetic Algorithm.

2.3.1 Genetic Algorithm Class Schedule

Class schedule is considered as a nondeterministic polynomial (NP) problem [19]. This problem can be solved by a few algorithms such as heuristic algorithm to find the optimal solution which only works for simple cases. The timetable of the class schedule is generating by using Genetic Algorithm due to its complex input and requirement. Genetic Algorithm only implements the hard requirements which can cause the schedule infeasible if break one of them. Soft requirements such as time and classroom preferred by professor are not implemented and it would not affect the schedule and the schedule is still feasible. There are the hard requirements for generating a class schedule:

Hard Requirements:

- The class can be placed only in a spare classroom.
- No professor or student group can have more than one class at one time.
- The classroom must have enough seats to accommodate all students.
- To place a class in classroom, the classroom must have laboratory equipment (computer in your case) if the class requires it.

(i) Objects of Class Schedule

There are five objects that make a class schedule which are professor, student group, course, class and classroom. The Professor class contains ID, name and all the classes that teach by the professor. The StudentsGroup class contains ID, name of 18 the group, size of the group, and a list of classes that the group attends. The Classroom class contains ID, name of the classroom, seats in the classroom and equipment of the classroom. The Course class holds information about ID and name of the course. The CourseClass class which is an important class holds a reference to the course to which the class belongs, reference to the professor who teaches the class, and a list of student groups that attend the class. It also stores how many seats are needed in the classroom, if the class requires computers in classroom and the duration of class (in hours). All of these objects are crucial. It will help to configure the Configure File which will discuss later.

(ii) Chromosome Representation

When deal with a Genetic Algorithm, the first things that have to consider is how to represent the solution in such a way that it is feasible for genetic operations such as crossover and mutation. Thus, it also needs to calculate the fitness value of our solution. To represent the chromosome for a class schedule, there are some assumption that made by this application. It assume that time is in one hour granules in the time-space slot for everyday and every room. It also assumes that class cannot begin before 9am and no class after 9pm. Total available hours will be 12 hours. Class will be start from Monday until Friday. It assumes working day is 5 days. This application use std::vector with size 2*5*number_of_rooms.

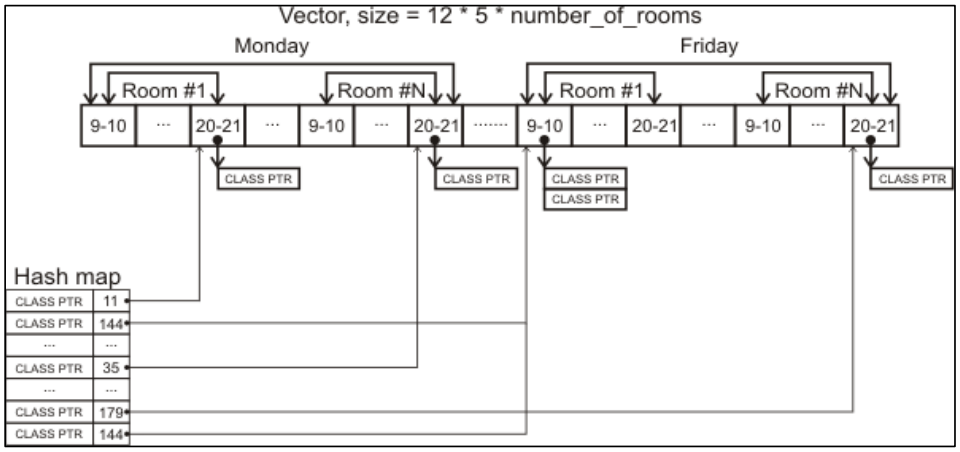


Figure 2.7: Chromosome Representation. There is an additional hash map that is used to obtain the first time-space slot at which class begins (its position in vector) from the address of a class's object. Each hour of a class has a separate entry in the vector, but there is only one entry per class in the hash map.

(iii) Fitness

Every chromosome need to assign its fitness. Thus, only hard requirements are used to calculate the fitness of chromosome for a class schedule. There are the calculations that use in the class schedule:

- Each class can have from 0 to 5 points.
- If the class uses a spare classroom, you increment its score.
- If the class requires computers and they are located in the classroom, or it doesn't require them, you increment the score of the class.
- If the class is located in a classroom with enough available seats, guess what, you increment its score.
- If the professor has no other classes at the time, you increment the class's score once again.

- The last thing that you check is whether any of the student groups that attend class has no other classes at same time and, if they don't, you increment the score of the class.
- If the class breaks a rule at any time-space slot that it occupies, its score is not incremented for that rule.
- The total score of the class schedule is the sum of points of all classes.
- The fitness value is calculated as schedule_score / maximum_score; the maximum_score is number_of_classes*5.

(iv) Crossover

In the hash map, two parents are selected to combine data. The crossover points are set and data will be exchange to produce new chromosome. The following picture show that the blue color data will combine into the red color slot. The white color slots which located between the red colors slots will be exchange with the blue color slot thus produce a new vector of slots.

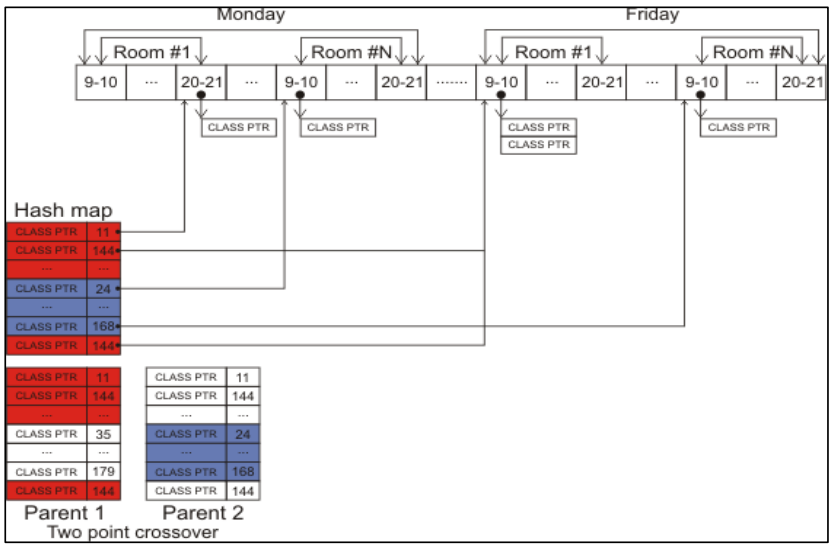


Figure 2.8: Crossover of Genetic Algorithm Scheduler

(v) Mutation

A mutation operation is very simple which is just randomly taking a class and moves it to another randomly chosen slot. The mutation size in the chromosome's parameters will define the number of classes which are going to be moved in a single operation.

(vi) Algorithm

There are two basic operation will be performed in each generation for Genetic Algorithm.

(i) Randomly selects N pairs of parents from the current population and produces N new chromosomes by performing a crossover operation on each pair of parents.

(ii) Randomly selects N chromosomes from the current population and replaces them with new ones. The algorithm doesn't select chromosomes for the replacement if it is among the best chromosomes in the population.

These two operations will be repeated until the fitness value become 1. Genetic algorithm keeps track of the M best chromosomes in the population and guarantees that they are not going to be replaced while they are among the best chromosomes.

(vii) Configuration

The Genetic Algorithm Class Schedule need configuration file to run its application. Each object in the configuration file begins with its tag and terminated with *#end* tag. There are the types of objects for the Genetic Algorithm Class Schedule:

- professor (#prof tag) - describes a professor

- course (#course tag) - describes a course
- room (#room tag) - describes a room
- group (#group tag) - describes a student's group
- course class (#class tag) - describes a class, and binds the professor, course, and students group.

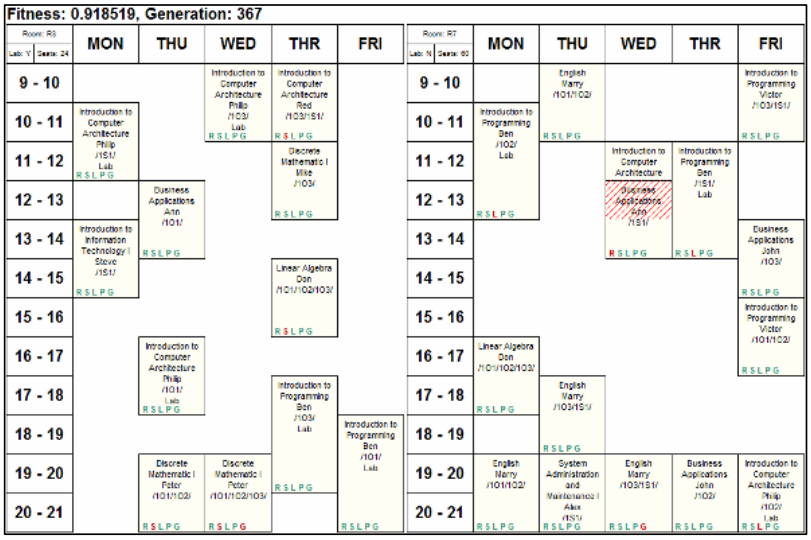


Figure 2.9: Interface of Genetic Algorithm Schedule with its Fitness Value

2.3.2 Faculty Timetabling using Genetic Algorithm

The Faculty Timetabling using Genetic Algorithm (FTGA) was developed by Boon Yaun Liong to generate a timetable for faculty use [20]. The main goal of this application is to produce an optimum timetable based on the availability of the classes at each time for each course of faculty.

There are only involved hard constraints in developing this application without cover the soft constraints due to it do not affect the feasibility of the timetable. The hard constraints that were used in FTGA are shown at below.

- If the class located in the classroom that has enough seats, the score will be increment.
- If the class is lab session and it located in the laboratory, the score will be increment.
- If the lecturer has no other classes at the same time, the score will be increment.
- If the student group has no other classes at the same time, the score will be increment.
- If no class between 13.00 until 14.00, the score will be increment.
- If there is no class between 12.00 until 15.00 on Friday, the score will be increment.
- Some courses must take place in certain classroom.
- Some lecturers are not available for some certain period.

Figure 2.10: List of Hard Constraints

There are the Genetic Algorithm processes which include Encoding, Evaluation of a Timetable, Selection, Crossover and Mutation involved in the development of FTGA application.

(i) Encoding

The population for this application are assumed as the solution which formed by all the timetable of room. The room timetable which representing 'chromosome' in Genetic Algorithm is formed by classes. A room timetable contain of 42 timeslots which filled by the class slots. A class slot represent 'genes' in Genetic Algorithm is consist of lecturer, subject's name and student section, duration of the class and type of delivery.

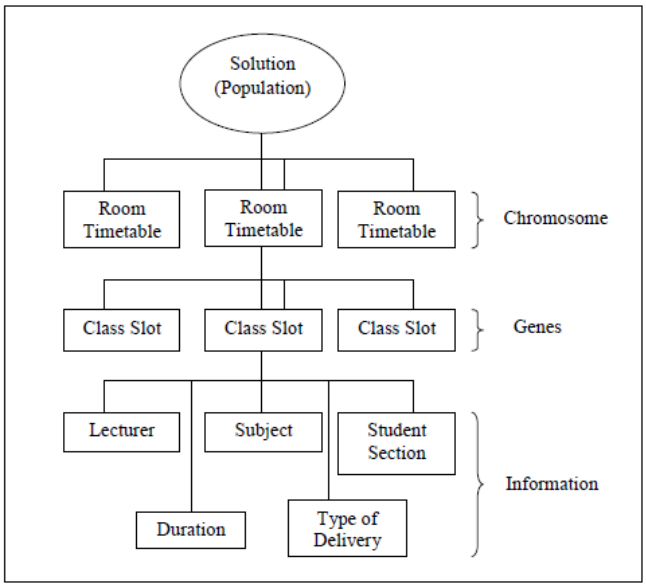


Figure 2.11: The Relationship between Chromosomes and Genetics for FTGA

(ii) Evaluation of a Timetable

The fitness for FTGA is representing by score. Fitness value is measure and calculated based on its constraints. The score for class slot will be incremented if the class slot is match with the requirement of constraints. The following are constraints which based on the requirement of faculty that will incremented the score:

- if the class located in the classroom that has enough seats
- if the class is lab session and located in the laboratory
- if the lecturer has no other classes at the same time
- if the student group has no other classes at the same time
- if there is no class between 13.00 until 14.00 (lunch time break) or between 12.00 until 15.00 on Friday

The class slot in the timetable will result the maximum score of the class slot for each room. Thus, the total fitness value of the population was summed up from the total fitness point of all the timetables.

(iii) Selection

FTGA application is performing elitist selection in producing new generation. After the evaluation from the first generation, the population with highest fitness value will be selected to enter the new generation and avoid it from crossover and mutation operation.

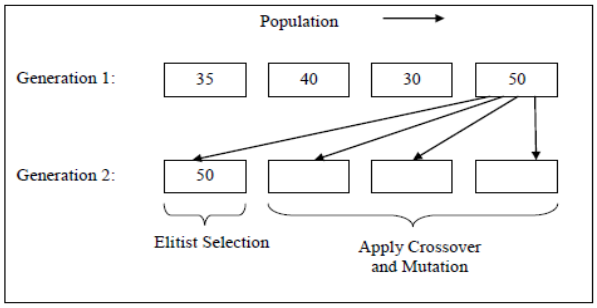


Figure 2.12: Elitist Selection of FTGA. The figure showing that the class slot which does not fulfilled the constraint will be selected to applying crossover process to obtain higher fitness point.

(iv) Crossover

The crossover operation in FTGA was performed between the selected class slot and an empty slot which is different from the normal crossover that crossover between two chosen class slot.

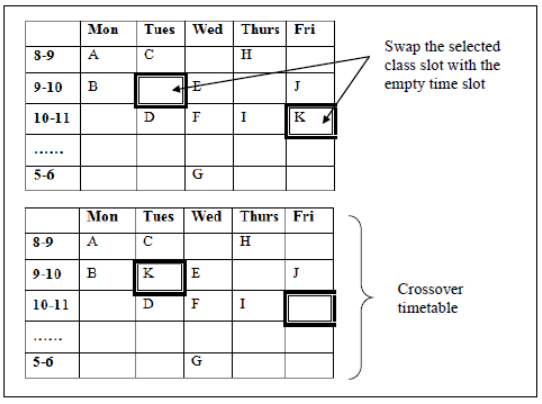


Figure 2.13: Illustration of Crossover Process for FTGA

(v) Mutation

In the application of FTGA, it is randomly select the class slot to applying mutation operation. The mutation process of FTGA was done in low rate (5%) which was not too high and avoids the delaying in obtaining the optimum solution.

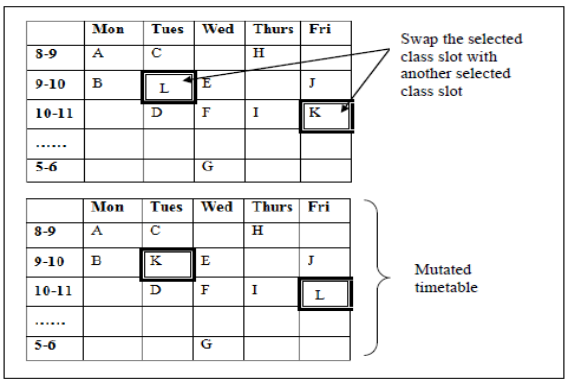


Figure 2.14: Illustration of Mutation Process for FTGA

2.4 Conclusion

Based on the literatures and researches that had been reviewed in this chapter, scheduling was consider as nondeterministic polynomial (NP) problem that contain of hard and soft constraints which have to consider and fulfill with its requirements by scheduler in producing a schedule. Besides that, scheduling problem is uncertain and changeable over the time. There are three main factors that concerning to scheduling problem which are environment of scheduling, scheduling act and the human resource affecting the scheduling. Thus, scheduling was a time consumption task due to it has a number of constraints and apprehension.

By study on the previous several literatures and researches, Genetic Algorithm had found to be a suitable technique that used to solve the optimization problem and obtain an optimum solution for scheduling. The operations of Genetic Algorithm such as reproduction, crossover and mutation are performed on the population and evaluated

each individual's fitness to obtain the fittest chromosome in the population. This means that Genetic Algorithm capable to obtain the best and optimal solution for the problem by performing its operation. Thus, an optimum schedule which able to satisfy all of its constraints can be generate by implementing with Genetic Algorithm technique.

As a conclusion, Genetic Algorithm is suitable to use in developing this project which is to produce an optimum PSM presentation timetable. Genetic Algorithm was trust to be able in solving the problem statements and constraints of the PSM presentation scheduling that have been discussed in Chapter 1.

CHAPTER 3

METHODOLOGY

This chapter will devoted to discuss the software process model which including the planning, analysis, design and development phases. The hardware and software specification that required for this project also will be discussed in this chapter.

3.1 Development Model

A software process model had planned as a system development method for this project. This model consist of 9 phases which is starts from planning, followed by requirements gathering, analysis, design, build prototype, customer evolution, refine requirements, development and testing. The 4 phases of design, build prototype, customer evolution and refine requirements is a cycling process which means that a prototype is built, tested and refine as necessary until the prototype is acceptable by customer and the complete system can now be developed. The flow of the software process model had shown at below.

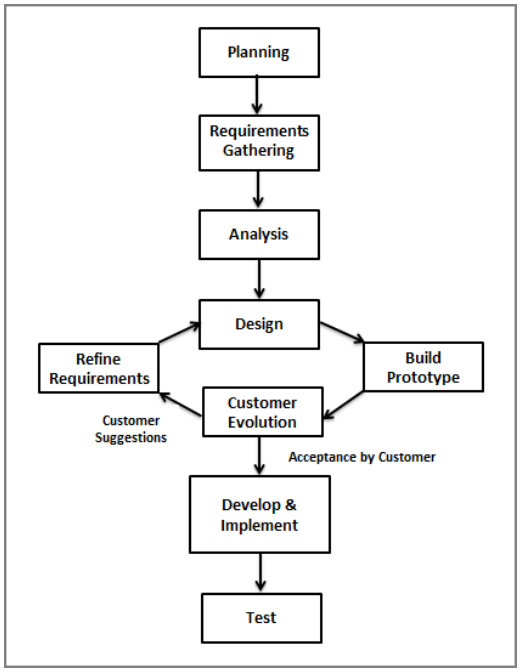


Figure 3.1: Flow of Model

3.1.1 Justification of Planning Development Model

This model is planned based on the environment and requirements that needed for the project. Assignation of PSM Evaluator using Genetic Algorithm (APEGA) is a project that involve with customer due to it is specific develop a schedule for a university's faculty use. Thus, this model that had planned is the illustration of input data formats, messages, reports and the interactive dialogs to the customer. Thus, it will be the way for gaining better understanding of the customer's needs due to prototyping requires customer involvement and allow them to provide feedback and specifications. The customer's requirement and feedback is the main issues that use in the system development. Therefore, the output of the project will be the most likely with the customer's needs.

Requirements gathering will be the next phase after the model planning. The purpose of this phase is to collect the basic information and requirements which is relevant with the project from the customer. In this phase, the initial requirements will

be collected from customers based on his request and requirements on the scheduling system for final year project presentation. The initial requirement and information from the customer is crucial to provide some ideas for the prototype design. The requirements will be keeping update with the customer to ensure that the prototype built does not have great differences with customer's needs.

The process of analysis is need after the initial requirements were collected from customer. Analysis phase is an essential process that aims to understand and analyze customer's needs and problems. Thus, the customer's requirements can be analyzed from the current practice and problem statements that had faced by the customer. Besides of this, customer will request some of the new requirements to be add-on in the system for as an enhancement from current system. All of the customer's requirements will be analyze in order to plan and design a prototype that based on it.

A cycling process which involved the phases of quick design, build prototype, customers evaluation and refine requirements will be the next procedure after the analysis phase. The purpose of these 4 phases is to ensure a prototype of the system is generated and satisfy with the customer's needs. When a prototype is designed and built based on the customer's requirements, it will be deliver to the customer for the evaluation purpose. The customer is responsible to evaluate on the prototype and request for change based on the evaluation and customer's suggestions. Then the prototype will be review and update based on customer's requirements. Thus, all of these 4 phases will keep repeating until the customer is satisfied with his desire prototype design.

When the final prototype is acceptance by customer and completely done, it is now come to the development phase. Development is the phase that discussing the steps about how the system will be construct by implementing the specific technique. In this phase, he prototype will be develop and implement with the specific technique to become a real system. The development and implementation process is involved with the aid of some hardware and software. All the software and hardware specification that had been used during the system development stage will be discussed in the topic of development and implementation.

Testing phase is the last step of the model which it is need for system testing. It is a crucial process that after system development to ensure that the system is run in free error and bugs before delivery it to the customer.

3.2 Planning

This phase is required to determine the feasibility of a particular project proceeding. In this phase, it will discuss about the flows of planning for the project including consultation with supervisor, choosing project title, define the problem statements, objectives and scopes. Thus, this phase will produce high level overview document of the proposed project.

3.2.1 Consultation with Supervisor

In the initial stage for planning this project, there are few discussion have been done with Miss Zalili Binti Musa to discuss about the project‘s title that will going to be proposed. Besides that, supervisor will consult about how the way to propose the project and what technique that will be used to implement the project.

3.2.2 Choosing Project Title

The project’s title had been chosen after the consultation with supervisor. The project’s title that will going to propose is “Assignation of PSM Evaluator using Genetic Algorithm (APEGA)”.

3.2.3 Definition of Problem Statements

The problem statements of this project are defined based on the problems that had faced by the current practise for presentation scheduling. The purpose for

developing this project is based on the needs and importance of this project which can help to solve the current problems.

3.2.4 Definition of Research Objectives and Project Scopes

Once the problem statements had been defined, the objectives and scopes of the project was stated in order to produce a guideline for searching more information which relevant to this project. In other hand, the project's objectives and scopes will as a refer statement for the development of the whole project.

3.2.5 Planning for Project Milestone

Project milestone is an earlier stage of planning for the project which including all the relevant tasks that will be implement in this project. A Gantt chart will be constructs in the planning stage as an overview for the schedule of whole project development. (Refer to Appendix A)

3.3 Initial Requirements

Initial requirement is the basic information that collected from customer or client of the project which provide a guideline for the prototype design based on the objective of the project. The objective of the Assignation of PSM Evaluator using Genetic Algorithm (APEGA) system is aim to assign the optimum PSM evaluators for the students and use of Faculty of System Computer and Software Engineering (FSKKP), University Malaysia Pahang during PSM presentation carnival.

3.3.1 Meeting with Client

To develop the system of APEGA, it is need to have the aid of a client who are experienced and have knowledge about scheduling. Thus, all the required information

such as customer's requirements, scheduling's constraints and target of the project can be obtained from the client of this project.

The client for this project is Encik Rahiwan Bin Romli who is PSM Coordinator for FSKKP. Encik Rahiwan Bin Romli has much experience in managing the final year project's stuffs which including prepare the schedule for PSM presentation. There are few interview sessions had been made with the Encik Rahiwan and he had stated out some of the requirements and constraints for matching PSM evaluator. All of the requirements that likely to contain in the system are recorded.

3.3.2 Customer's Requirements and Constraints

The requirements that had stated by the client are based on the problems and constraints from the current practice for assignation of PSM Evaluator. As stated, the requirements of the APEGA are listed as below:

i. The system can auto assign the PSM evaluator for the student instead of the manually matching.

ii. All of the student's projects and evaluator's expertise are categories in four main fields which are
    - Computer Science (CS) (including Information System or Management Information System)
    - Computer System and Networking (SN)
    - Software Engineering (SE)
    - Graphic and Multimedia (GMM)

iii. The main constraints for the assignation of PSM Evaluator are:
    - Each student's project and presentation must be evaluated by 2 evaluators.
    - Among the two evaluators, one of the evaluator must have the relevant expertise field based on the student's project and another one of the evaluator can be randomly chosen from any field of lecturer.

- All the evaluators are not allowed to evaluate the students who are under their supervision.

Besides that, there are some data such like the detail information about student's project and evaluator for PSM presentation had collected from Encik Rahiwan Bin Romli as the reference for develop this project.

3.3.3 Specification Requirements

There are several software and hardware have been used to developing the system of Assignation of PSM Evaluator using Genetic Algorithm (APEGA). Hardware is the physical aspect of the computer while software is the general part for the various kinds of program used to operate computer and other devices. The following section will show the detail of hardware and software specifications that use for the system development.

3.3.3.1 Software Specification

The table below shows the list of software that used to developing the system.

Type of Software	Name	Purpose
Operating System	• Microsoft Windows Vista™ Home Premium	➢ To run the computer program
Software Application	• Microsoft Visual Studio 2010	➢ To develop the PPSGA (Coding).
	• Microsoft SQL Server 2005	➢ To store data that used by the system
	• Microsoft Office 2010 - Microsoft Office Word 2010 - Microsoft Office Project 2010 - Microsoft Office PowerPoint 2010 - Microsoft Office Visio 2010	➢ For documentation, to generate presentation slide, Gantt Chart, and graphical diagram.
	• Adobe Reader 9	➢ To read the file in pdf format
	• WinRAR 3.93	➢ To compress and decompress file

Table 3.1: Software Specification for APEGA

3.3.3.2 Hardware Specification

The table below shows the list of hardware that is used during the development of the system.

Hardware	Specification	Purpose
Laptop	• Intel(R)Core(TM) Duo CPU P7350 @ 2.00GHz • 4 GB RAM • 320 GB Hard Disk	➢ To run the operating system for development
Printer	• Canon Pixma MP145	➢ To print out the documentation
External Hard Disk	• No Touch 600 GB	➢ Backup and transferring data
Compact Disk	• 700 MB 48x	➢ System backup and delivery

Table 3.2: Hardware Specification for APEGA

3.4 System Analysis

At the earlier stage of analysis, literature review had been done for the purpose to get more knowledge that relevant with the project's title. Few journal and research paper that about scheduling and Genetic Algorithm had been review and discussed in chapter 2. By reviewed on the existing systems, the relevant information can be analysed and help to enhance more understanding about the project.

System analysis is a stage that analyse the needs of the customer which have to add-on into the system and also the system's functionality. An analysis will be done based on the initial requirements that had collected from the client of this project, Encik Rahiwan Bin Romli. All the basic requirements, scheduling constraints, and information will be analyse and arrange in order to produce a guideline for prototype design. Therefore, the analysis process has to start from the problem statements due to some of the customer's requirements are gaining from the current practice and problems that had faced by the client.

3.4.1 Analysis Based on Customer's Requirements

Based on the initial requirements that had provided by Encik Rahiwan Bin Romli, he expected that the scopes of this APEGA system is able to generate an presentation schedule that cover minimum 200 of students project and 40 evaluators. All of the student's projects and evaluator's expertise are categories in four main fields which are Computer Science, Computer System and Networking, Software Engineering, Graphic and Multimedia.

3.4.2 Analysis on Implementation Technique

By referring to the constraints and customer's requirements, Genetic Algorithm technique is chosen to use in solving these scheduling optimization problems. Thus, APEGA is the system that develops and implements with Genetic Algorithm in order to generate an optimum list of the distribution between student's project and their evaluators. The detail procedures and explanation for the prototype design will be discussed in the next topic.

3.5 Design

Design is the phase that after the system analysis had been done and the functionality of the system is set. In this stage, the interaction with the customers is important in order to build and design the models or prototype for the system.

3.5.1 System Design

The purpose of this phase is to design the flow and the way that how the system running. The general flow of the APEGA system is show at the figure below.

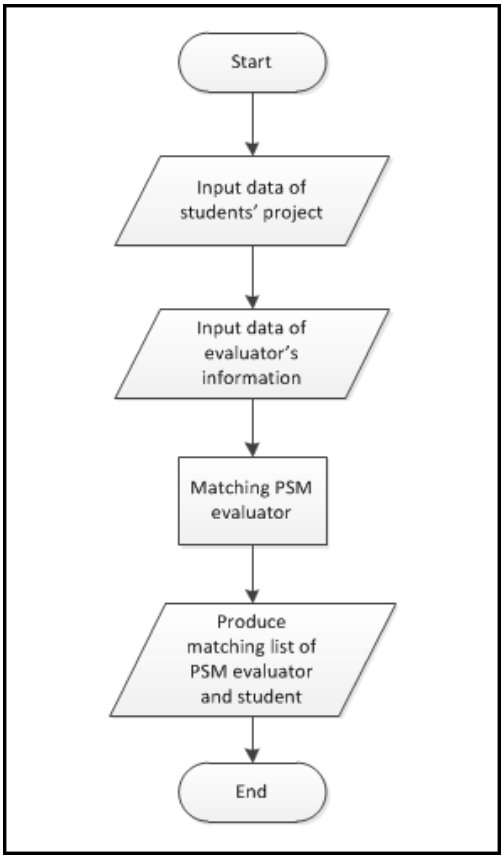


Figure 3.2: Flow Chart of APEGA System

The APEGA system is start by the user, PSM coordinator who is responsible in assigning PSM evaluator. The PSM coordinator has to insert the few data which are the detail of students' project and the information of evaluators. Then, the system will auto rearrange the input data to become a systematic output data. In the following step, the system will use the output data to matching the optimum evaluators for the students with the Genetic Algorithm concept. The system will end when a matching list of PSM evaluator and student is produce.

3.5.1.1 Input Data of System

The information such like detail of student's project and evaluators are the main and essential data that use to matching the optimum evaluators for the students. Thus, all of these data have to identify clearly with its categories. This means that the information of the evaluators must including their expertise field and the student's project must be stated clearly with its category.

i. The basic data of student's project that are needed for generate presentation schedule are:
    - Student Name
    - Matric No
    - Supervisor
    - Project Title
    - Project Category

ii. The basic data of evaluators that are needed for generate presentation schedule are:
    - Lecturer Name
    - Field of Expertise (can be more than one)

3.5.1.2 Output Data of System

The APEGA system then will retrieve the data of evaluators from the database and randomly distribute 2 evaluators in 1 group. Therefore, it is assume that will have 6 groups of evaluator from the division of 12 lecturers which had stated at the project scope in chapter 1. Thus, each of the evaluator is assumed will evaluate 5 students at average. When the student's list that stated above is generated, the system will randomly assign students into each group of the evaluators. The assignation between students and evaluators will be named as "Evaluator Slot" which will be discussed in the next topic.

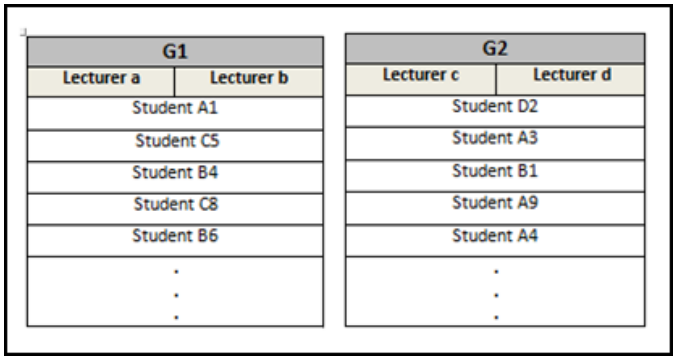


Figure 3.3: Example of Evaluator Slot

3.5.1.3 Assumption of System

Besides that, there are some assumption that made by this system. It assumes that the number of student in each evaluator slot will not be same at all due to it is randomly distributed by the system.

3.5.2 Software Design

The implementation of Genetic Algorithm into APEGA will be explained in this phase. The Genetic Algorithm processes are including Encoding, Fitness Evaluation, Crossover and Mutation.

3.5.2.1 Encoding

The technique of Genetic Algorithm is implemented to solve the optimization problem for the distribution in between the students and their evaluator where an evaluator must evaluate the project which is regarding to his or her field or expertise. Encoding is a process that determines the chromosome representation in Genetic Algorithm. The list of the elements of Genetic Algorithm based on the APEGA system are shown as below:

i. **Population** = PSM presentation schedule

ii. **Chromosome** = Evaluator slot (The assignation of students for each group of evaluator.)

iii. **Gene** = Information slot (Evaluator expertise & Students' Project)

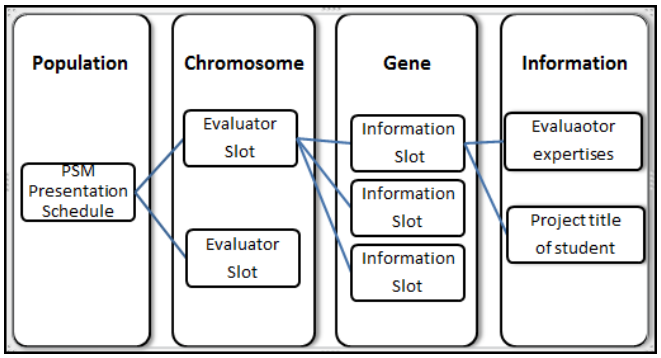


Figure 3.4: Relationships between Elements of Genetic Algorithm

Based on the figure above, the schedule for PSM Presentation represents the population in Genetic Algorithm concept which consists of a number of chromosomes. The chromosomes are represented by evaluator slot which is formed by a number of information slots. Each of the information slots which as the gene in Genetic Algorithm are formed by referring to the information of evaluator's expertise and student's project.

Information slot is a formation of the assignation of students and their evaluators. All the information slots will combined in order to produce evaluator slot. All of these evaluator slots will be evaluated by Genetic Algorithm operation to obtain the highest fitness value for them. Thus, each of the evaluator slots that with highest fitness value will be arrange into the presentation times slot in sequence. The detail of Genetic Algorithm operation based on APEGA system will be discussed in the following topic.

3.5.2.2 Fitness Evaluation

Every chromosome need to assign its fitness. The fitness of the schedule is based on the constraints and it is no specific function to calculate it. The fitness value is

represented by score in the condition of if the evaluator slot is fulfills with the constraints, and then the score will be incremented [21].

- The point of the evaluator slot is set from 0.
- If the evaluator who expert in Software Engineering evaluate the project about System Information or Artificial Intelligent, the score will be increment.
- If the evaluator who expert in Networking evaluate the project about System Security or Network, the score will be increment.
- If the evaluator who expert in Multimedia and Graphic evaluate the project about Image Processing, Courseware or Modeling the score will be increment.
- If the evaluator evaluate the student who not under his or her supervision, the score will be increment.
- The total score of the lecturer slot is the sum of points of two evaluators.

Group 1		
	Lecturer a	Lecturer b
Student B1	1	0
Student D5	0	1
Student A4	1	1
Student B8	1	0
Student C6	0	0
TOTAL	3	2
	5	

Figure 3.5: Evaluation of Fitness Values

The maximum score of the evaluators which means the two evaluators with the highest of sum of fitness value in the evaluator slot will be choose to evaluator for the related student. The evaluator slot with highest fitness value will be chosen to proceed in next generation. However, the evaluator slot with lowest score will be auto discard from the selection list and chosen to perform crossover operation.

3.5.2.3 Crossover

The crossover operation for APEGA system is select two evaluator slots (chromosomes) which does not fulfills with the constraints or having lowest fitness value. Thus, randomly choose one of the students from each evaluator slots and swap between them. The purpose of crossover operation is to change over the assignation of students in order to produce new evaluator slot with higher fitness value.

The steps for the crossover operation are shown as below:

Step 1: Recognize two of the evaluator slots which having the lowest fitness value.

Step 2: Randomly selected one of the students from the list for each evaluator slot.

Step 3: Swap between two selected students.

Step 4: Two new evaluator slots are produced.

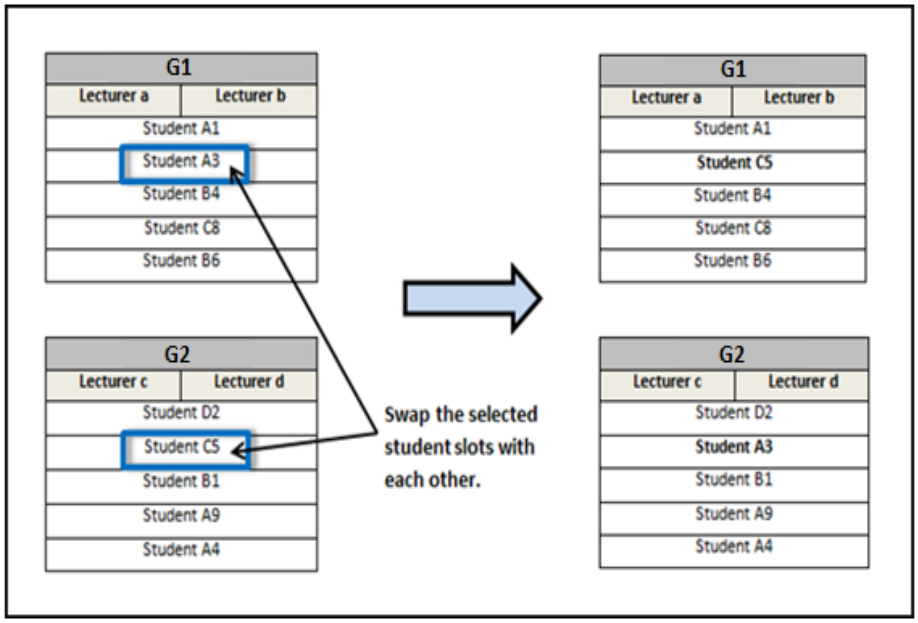


Figure 3.6: Illustration of Crossover

3.5.2.4 Mutation

For the mutation of APEGA system, some of the evaluator slots will be randomly selected to perform mutation operation and change its student's assignation. In this project, the mutation process should be done in low rate due to the process to approach the optimum solution will slow down if the mutation was done it high rate. The purpose of mutation operation is to change the student from the evaluator list which might obtain higher fitness values.

The steps for mutation operation are shown as below:

Step 1: Randomly choose one evaluator slots.
Step 2: Change the student from the list of evaluator slots.
Step 3: A mutated evaluator slot is produced.

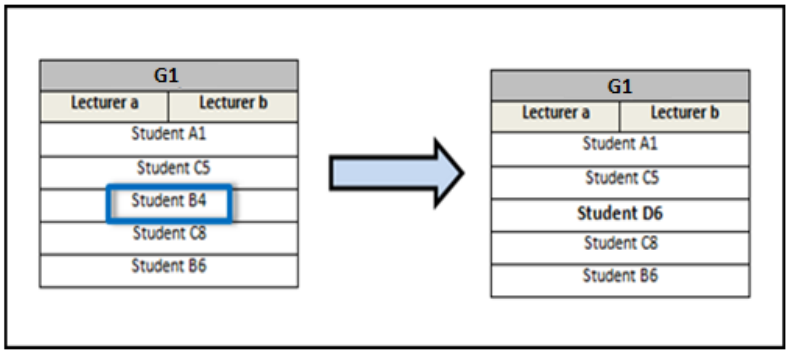


Figure 3.7: Illustration of Mutation

3.6 Prototype

A prototype will be constructed when the sketch of the system is design regarding to the customer's requirements. The development of prototype is including design, coding and testing. But each of these phases will be done without formally as it just to test the functionality before the development of real system. Thus, the scope for the prototype is exactly smaller than the actual APEGA system.

In this phase, the main constraint of this system which is the assignation of the students for each evaluator according to their field or expertise will be implemented. The others constraints will not applied in the prototype yet. The purpose of this phase is enable customer to interact with the prototype for better understanding the operation of the desired system.

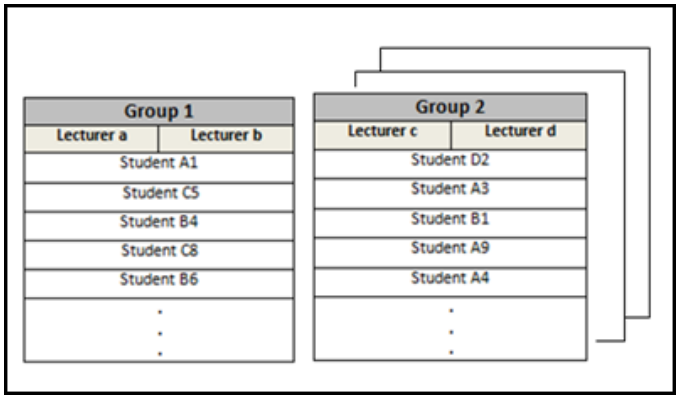


Figure 3.9: Prototype of A Set of Evaluator Slot that generated by APEGA

3.7 Customer Evaluation

Customer evaluation is a stage that delivered the prototype to the customer for evaluation after it is developed. The customer of this project, Encik Rahiwan Nazar Bin Romli will evaluate the prototype based on his requirements and he is responsible to request for the change of the prototype.

3.8 Refine Requirement and Update

After the evaluation of prototype, the changes and requirements that request by customer, Encik Rahiwan Nazar Bin Romli will be review and apply in devloping new prototype. The prototype will be repeating change and modify until it is approach and achieve with customer's needs. The final prototype which is exactly fulfilled with customer's requirements will be design after the customer's confirmation.

3.9 Development and Implementation

Development phase starts when the final prototype is designed. In this stage, the prototype will be developed formally. All the constraints are used to evaluate the fitness of the schedule. A list of students' name and their evaluators will be generated as like evaluator slot that had stated at previous section. The full system for Assignation of PSM Evaluator is developed by using Genetic Algorithm technique that had been explained in system and software design phase. The detail of procedures in this phase will be explained in chapter 4.

3.10 Testing

After the code is generated and system development is done, the system will be testing to ensure it is free from error and bug. In this phase, the functionality of the system will be test to ensure that it is able to matching the assignation of PSM evaluator for the student with the condition of fulfills all the constraints that had been stated at previous section. The detail of testing part will be discussed in chapter 5.

CHAPTER 4

IMPLEMENTATION & RESULT

This chapter will be devoted for the discussion on the implementation of Genetic Algorithm technique in the system.

4.1 Implementation

In the implementation of APEGA system, it will involve with few operations and steps of Genetic Algorithm. The system implementation stage will start by generate random population and follow by fitness value calculation, Elitism Selection, crossover and mutation. The detail of each step will be discussed at the following topics.

4.1.1 Generate Random Population

The earliest stage for the Genetic Algorithm is to generate random population. In this system, there are only 3 population will be generated. The students' matric number will be randomly assigned into any group of two evaluators which known as evaluator slot in this project. The output for the system will be the list of each evaluator slot together with the students' matric number which had been randomly assigned into their group. Each group of evaluator will represent by a table where the rows are represented by row and the column is represented by coulmnTable**.** The number of evaluator slot will be represented by totalSlot. Each of the students' matric number will be randomly placed in the evaluator slot by assigning 3 values which are row, columnTable, and totalSlot. These three values represented which evaluator slot that the student had been assigned.

The evaluator slot must be empty before the students' matric number is placed into it. The checkRmpty SqlCommand is use to check whether the evaluator slot is empty and the result will be represented by availability. If the availability is 0 means that the evaluator slot is empty where the row, columnTable, and totalSlot have not assigned by any students' matric number yet.

```
Dim row As Integer, column As Integer, noEvaluatorSlot As Integer = 1
  For noEvaluatorSlot = 1 To 2
    For row = 1 To 5
      For column = 1 To 1
'Check Evaluator Slot availability
Dim checkEmpty As New SqlCommand("SELECT COUNT(*) FROM Data WHERE population = '" &
population & "' AND totalSlot = '" & noEvaluatorSlot & "' AND row = '" & row & "' AND columnTable = '"
& column & "'", con1)
Dim availability As Integer = checkEmpty.ExecuteScalar()
```

Figure 4.1: Coding for Check Evaluator Slot Availability

When the evaluator slot had been checked and it is empty, a RandomNumber will be picked between 0 until 8. If the RandomNumber is below 6, a student will be randomly select from the StudentInfo table. Once the student had been selected, the other information for the relevant student such as matric number, supervisor's name, category of project will be retrieved from the StudentInfo table and insert into the Data table. All the information of generate random population will be inserted into the database (refer to Appendix B).

```
Dim RandomStudent as New Random()
Dim RandomNumber as Integer
RandomNumber = RandomStudent.Next(0, 8)
 If RandomNumber < = 6 Then
    con.Open()
Dim randomSelect As New SqlCommand("SELECT TOP 1 studentNo FROM Student_Info1 WHERE count =
'1' ORDER BY NEWID()", con)
Dim randomStudent As String = randomSelect.ExecuteScalar()
randomSelect.Dispose()
```

Figure 4.2: Coding for Generate Random Population

```
con.Open()
Dim cmdschedule As New SqlCommand("SELECT Matric_No FROM Student_Info1 WHERE count = '1'
AND studentNo='" & random1 & "' ", con3)
Dim studentMatric As String = Convert.ToString(cmdschedule.ExecuteScalar())
con.Close()

con.Open()
Dim getProjectType As New SqlCommand("SELECT Project_Cat FROM Student_Info1 WHERE count = '1'
AND studentNo='" & random1 & "' ", con)
Dim projectType As String = getProjectType.ExecuteScalar()
getProjectType.Dispose()
con.Close()

con.Open()
Dim getSV As New SqlCommand("SELECT Supervisor FROM Student_Info1 WHERE count = '1' AND
studentNo='" & random1 & "' ", con)
Dim SV As String = getSV.ExecuteScalar()
getSV.Dispose()
con.Close()
```

Figure 4.3: Coding for Retrieve Data from StudentInfo Table

4.1.2 Fitness Value Calculation Function

After the population had been randomly generated, the fitness value for each population will be calculated. The fitness value is calculated based on the constraints which had discussed in chapter 3. The score for the fitness value will be incremented if it is fulfill the constraints as below:

- If the evaluator who expert in Software Engineering evaluate the project about System Information or Artificial Intelligent, the score will be increment.
- If the evaluator who expert in Networking evaluate the project about System Security or Network, the score will be increment.
- If the evaluator who expert in Multimedia and Graphic evaluate the project about Image Processing, Courseware or Modeling the score will be increment.
- If the evaluator evaluate the student who not under his or her supervision, the score will be increment.

In order to check the fulfillment of the constraints, each student in the evaluator slot will be go through one by one to calculate its fitness value. For the calculation of fitness value, the project's type of the student in each evaluator slot will be retrieve from the Data table and compare with the evaluator's expertise field. The following is the coding for retrieving the project's type of the student through each evaluator slot.

```
For population = 1 To 3
For noEvaluatorSlot = 1 To EvaluatorSlot
For column = 1 To 1
For row = 1 To 5
 fvLect1 = 0
Dim checkType As New SqlCommand("SELECT projectType FROM Data WHERE population = '" &
population & "' AND row = '" & row & "'AND columnTable = '" & column & "' AND totalSlot = '" &
noEvaluatorSlot & "'", con)
Type = Convert.ToString(checkType.ExecuteScalar())
```

Figure 4.4: Coding for Retrieve Current Project's type of the Student

The following step is to get the both evaluators' expertise (E_{i_j}) for the current group of evaluator from the GroupEvaluator table. As reference, 'E' is represent evaluator while '$_i$' is refer to number of evaluator in the evaluator slot and '$_j$' is refer to number of expertise of the evaluator '$_i$'. For example, the checkE1_1 command will retrieve the first expertise for the first evaluator in the group where Group_No is equal to current noEvaluatorSlot. The checkE2_1 command will retrieve the first expertise for the second evaluator in the group where Group_No is equal to current noEvaluatorSlot. Each evaluator had been assumed that have maximum three expertise field. Therefore, the coding below is to get all the expertise for both evaluators in the first group of evaluator from the GroupEvaluator table.

```
'check evaluator1 for group 1
'check evaluator1 – expertise 1
Dim checkE1_1 As New SqlCommand("SELECT E1_1 FROM Group_Evaluator WHERE Group_No = '" &
noEvaluatorSlot & "'", con)
e1_1 = Convert.ToString(checkE1_1.ExecuteScalar())
'check evaluator1 – expertise 2
Dim checkE1_2 As New SqlCommand("SELECT E1_2FROM Group_Evaluator WHERE Group_No = '" &
noEvaluatorSlot & "'", con)
e1_2 = Convert.ToString(checkE1_2.ExecuteScalar())
'check evaluator1 – expertise 3
Dim checkE1_3 As New SqlCommand("SELECT E1_3 FROM Group_Evaluator WHERE Group_No = '" &
noEvaluatorSlot & "'", con)
e1_3 = Convert.ToString(checkE1_3.ExecuteScalar())

'check evaluator2 for group 1
'check evaluator2– expertise 1
Dim checkE2_1 As New SqlCommand("SELECT E2_1 FROM Group_Evaluator WHERE Group_No = '" &
noEvaluatorSlot & "'", con)
E2_1 = Convert.ToString(checkE2_1.ExecuteScalar())
'check evaluator2 – expertise 2
Dim checkE2_2As New SqlCommand("SELECT E2_2FROM Group_Evaluator WHERE Group_No = '" &
noEvaluatorSlot & "'", con)
E2_2 = Convert.ToString(checkE2_2.ExecuteScalar())
'check evaluator2 – expertise 3
Dim checkE2_3 As New SqlCommand("SELECT E2_3 FROM Group_Evaluator WHERE Group_No = '" &
noEvaluatorSlot & "'", con)
E2_2 = Convert.ToString(checkE2_3.ExecuteScalar())
```

Figure 4.5: Coding for Retrieve Evaluator's expertise From Group Evaluator Table

To check the fulfillment of the constraints, compare the string between the student's project type (Type) and Evaluator's expertise (e1_1, e2_1,…..). If the return value is equal to 0, means the student's project type is match with either one of the evaluator's expertise. Therefore the fitness value for both of the evaluator, fvLect1 and fvLect2 in the current group will be incremented. The total fitness value for one evaluator slot, LectFV is the sum of the fitness value for the two evaluators who are in the same group. The sum of all the evaluator slot's fitness value will become the fitness value for the population and it will be inserted into database.

```
'comparison between student project type and the expertise for evaluator1
If String.Compare(Type, e1_1) = 0 Or String.Compare(Type, e1_2) = 0 Or
String.Compare(Type, e1_3) = 0 Then
fvLect1 = fvLect1 + 1
'comparison between student project type and the expertise for evaluator2
If String.Compare(Type, e2_1) = 0 Or String.Compare(Type, e2_2) = 0 Or
String.Compare(Type, e2_3) = 0 Then
fvLect2 = fvLect2 + 1
End If
End If
LectFV = fvLect1 + fvLect2
Dim cmdGroupFV As New SqlCommand("UPDATE Data SET groupFv = '" & LectFV & "'WHERE
population = '" & population & "' AND totalSlot = '" & noEvaluatorSlot & "' AND row = '" & row & "' AND
columnTable = '" & column & "'", con)
cmdGroupFV.ExecuteNonQuery()
```

Figure 4.6: Coding for Fitness Value Calculation between Student Project Type and Evaluator's expertise

4.1.3 Elitism Selection

Elitism Selection will select the population with the highest fitness value to pass to next generation. First, the population with the highest fitness value will be cloned from the Population table into a new table named TempPopulation. The population will be selected by using MAX (FitnessValue). Function MAX will select the higher value for the column FitnessValue. After cloned the population which having the highest fitness value, the data in the Population table will be deleted and the other data for the population with lower FitnessValue will be eliminated in the same time. Then, the data in the TempPopulation table will be passed back to the Population table again. Therefore, the Population table is now only containing the data for the population which having the highest fitness value and the Temp Population table will be deleted immediately.

```
Dim cloneData As New SqlCommand("SELECT TOP 1 * INTO
TempPopulation FROM Population WHERE FitnessValue = (SELECT
MAX(FitnessValue) FROM Population)", con)
cloneData.ExecuteNonQuery()
Dim deleteData As New SqlCommand("DELETE FROM Population",
con)
deleteData.ExecuteNonQuery()
Dim returnData As New SqlCommand("INSERT INTO Population
SELECT * FROM TempPopulation", con)
returnData.ExecuteNonQuery()
Dim deleteTempTable As New SqlCommand("DROP TABLE
TempPopulation", con)
deleteTempTable.ExecuteNonQuery()
```

Figure 4.7: Coding for Elitism Selection

The Population with highest fitness value will be assigned with a parameter name HighestFitnessValuePopulation and it will be used for the Crossover operation.

```
Dim elitistSelection As New SqlCommand("SELECT Population FROM
Population WHERE FitnessValue = (SELECT MAX(FitnessValue) FROM
Population)", con)
Dim HighestFitnessValuePopulation As Integer =
elitistSelection.ExecuteScalar()
```

Figure 4.8: Coding for Assign Highest Fitness Value

4.1.4 Crossover

Crossover operation will only be performed on the evaluator slot that does not fulfill the constraints or having 0 fitness value to change over the assignation of students in order to produce new evaluator slot with higher fitness value.

```
Dim randomStudent1 As New SqlCommand("SELECT TOP 1 studentNo FROM Temp WHERE fvLect1
= '0' AND fvLect2 = '0' ORDER BY NEWID()", con)
Dim studentNo1 As Integer = randomStudent1.ExecuteScalar()
If studentNo1 = 0 Then
  'Do Nothing
Else
Do
Dim randomSlot As Integer
randomSlot = RandomStudent.Next(1, totEvaluatorSlot)
Dim randomStudent2 As New SqlCommand("SELECT studentNo FROM Temp WHERE totalSlot = '"
& randomSlot & "' AND row = '" & Row1 & "'AND columnTable = '" & Column1 & "' ", con)
Dim studentNo2 As Integer = randomStudent2.ExecuteScalar()
Dim checkE1_1 As New SqlCommand("SELECT E1_1 FROM Group_Evaluator WHERE Group_No =
'" & randomSlot & "'", con)
e1_1 = Convert.ToString(checkE1_1.ExecuteScalar())
Dim checkE1_2 As New SqlCommand("SELECT E1_2 FROM Group_Evaluator WHERE Group_No =
'" & randomSlot & "'", con)
e1_2 = Convert.ToString(checkE1_2.ExecuteScalar())
Dim checkE1_3 As New SqlCommand("SELECT E1_2 FROM Group_Evaluator WHERE Group_No =
'" & randomSlot & "'", con)
e1_3 = Convert.ToString(checkE1_3.ExecuteScalar())
Dim checkE2_1 As New SqlCommand("SELECT E2_1 FROM Group_Evaluator WHERE Group_No =
'" & randomSlot & "'", con)
e2_1 = Convert.ToString(checkE2_1.ExecuteScalar())
Dim checkE2_2 As New SqlCommand("SELECT E2_2 FROM Group_Evaluator WHERE Group_No =
'" & randomSlot & "'", con)
e2_2 = Convert.ToString(checkE2_2.ExecuteScalar())
Dim checkE2_3 As New SqlCommand("SELECT E2_3 FROM Group_Evaluator WHERE Group_No =
'" & randomSlot & "'", con)
e2_3 = Convert.ToString(checkE2_3.ExecuteScalar())
```

Figure 4.9: Coding for Retrieve Data that Have Low Fitness Value

During the fitness value calculation, the fitness value for the evaluator matching, fvLect will set to 0 if their expertises does not match with the student project's type. After the student who the fvLect = 0 and fvLect2 = 0 is selected, the system then will randomly choose another student to perform crossover operation. The crossover operation is about retrieve the current number of Evaluator slot, number of row and column for both students and then change over between the two students. Before the crossover operation, the porject type for both student will be retrieve and compare with the evaluator's expertise. If there are any matching among the project type, the fvLect will be increment by 1 and save to database.

```
If Not InStr(Type, e1_1) = 0 Or Not InStr(Type, e1_2) = 0 Or Not InStr(Type, e1_3) = 0 Then
Dim CrossoverNewSlot As New SqlCommand("UPDATE Temp SET totalSlot = '" & randomSlot & "',
row = '" & Row1 & "', columnTable = '" & Column1 & "', fvLect1 = '1' WHERE studentNo = '" &
studentNo1 & "'", con)
CrossoverNewSlot.ExecuteNonQuery()
If Not InStr(Type, e2_1) = 0 Or Not InStr(Type, e2_2) = 0 Or Not InStr(Type, e2_3) = 0 Then
Dim CrossoverNewSlot2 As New SqlCommand("UPDATE Temp SET totalSlot = '" & randomSlot & "',
row = '" & Row1 & "', columnTable = '" & Column1 & "', fvLect2 = '1' WHERE studentNo = '" &
studentNo1 & "'", con)
CrossoverNewSlot2.ExecuteNonQuery()
Else
    'Do Nothing
End If
Dim CrossoverSlot As New SqlCommand("UPDATE Temp SET totalSlot = '" & Slot1 & "', row = '" &
Row1 & "', columnTable = '" & Column1 & "' WHERE studentNo = '" & studentNo2 & "'", con)
CrossoverSlot.ExecuteNonQuery()
  'Exit Do
```

Figure 4.10: Coding for Crossover

4.1.5 Mutation

The mutation operation will be perform to change over the students between two selected evaluator slot and it will be done in low rate to avoid from slowing down the process of getting optimum solution. In this project, the mutation rate is set to 5%. The mutation rate is done by randomly select a number which is between 1 and 100. If the random selected number is less or equal to 5, the mutation operation will perform. Otherwise, the mutation operation will not perform if the random selected number is higher than 5.

```
Dim RandomClass As New Random()
Dim RandomNumber As Integer
RandomNumber = RandomClass.Next(0, 100)
If RandomNumber <= 5 Then
Mutation()
Else
'Do Nothing
End If
```

Figure 4.11: Coding for Set Mutation Rate

Mutation opertion is about randomly select two students who are from the population with highest fitness value to change over the number of Evaluator slot, number of row and column in order to produce the new fitness value of the matching between project type and the evaluator's expertises for both student. The differention of the mutation and crossover is the students which selected to perform mutation operation is choose by randomly and does not based on its fitness value. The system is using the SQL command, ORDER BY NEWID() to randomly select the student from database.

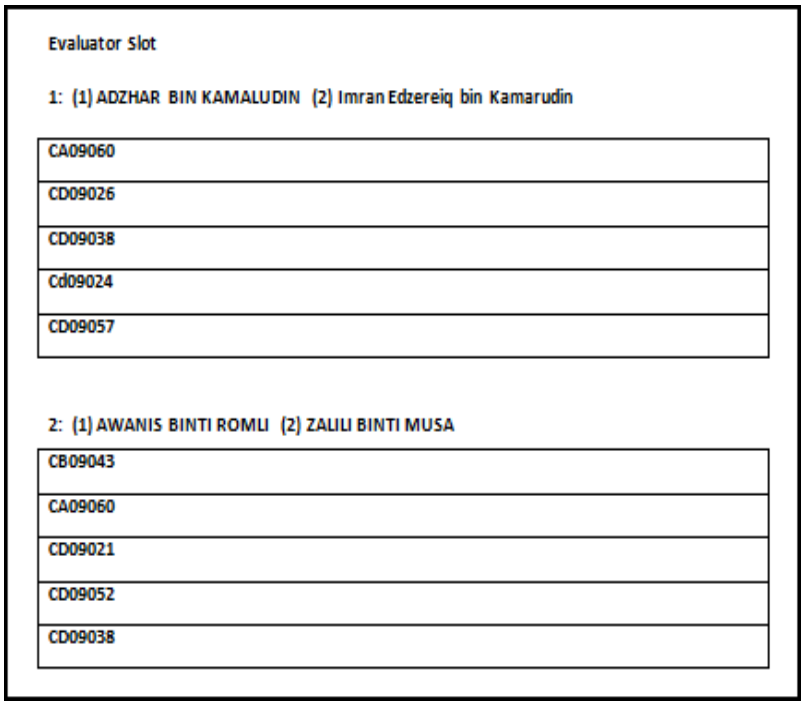
```
Dim population As Integer = HFVPopulation
Dim randomStudentSV1 As New SqlCommand("SELECT Top 1 studentNo FROM Data WHERE
population = '" & HFVPopulation & "' ORDER BY NEWID()", con)
 Dim firstStudentSV As Integer = randomStudentSV1.ExecuteScalar()
Dim randomStudentSV2 As New SqlCommand("SELECT Top 1 studentNo FROM Data WHERE
population = '" & HFVPopulation & "' ORDER BY NEWID()", con)
 Dim secondStudentSV As Integer = randomStudentSV2.ExecuteScalar()

'Get slot number, table row and table column
 Dim getNoOfSlot1 As New SqlCommand("SELECT totalSlot FROM Data WHERE studentNo = '" &
firstStudentSV & "' AND population = '" & HFVPopulation & "'", con)
Dim firstNoOfSlot As Integer = getNoOfSlot1.ExecuteScalar()
Dim getTableRow As New SqlCommand("SELECT row FROM Data WHERE studentNo = '" &
firstStudentSV & "' AND population = '" & HFVPopulation & "'", con)
Dim firstTableRow As String = getTableRow.ExecuteScalar()
Dim getTableColumn As New SqlCommand("SELECT columnTable FROM Data WHERE studentNo =
'" & firstStudentSV & "' AND population = '" & HFVPopulation & "'", con)
Dim firstTableColumn As Integer = getTableColumn.ExecuteScalar()
Dim mutation As New SqlCommand("Update Data SET totalSlot = '" & firstNoOfSlot & "', row = '" &
firstTableRow & "', columnTable = '" & firstTableColumn & "'WHERE studentNo = '" &
secondStudentSV & "' AND population = '" & HFVPopulation & "'", con)
mutation.ExecuteNonQuery()

'Get2nd slot number, table row and table column
Dim getNoOfSlot2 As New SqlCommand("SELECT totalSlot FROM Data WHERE studentNo = '" &
secondStudentSV & "' AND population = '" & HFVPopulation & "'", con)
Dim secondNoOfSlot As Integer = getNoOfSlot2.ExecuteScalar()
Dim getTableRow2 As New SqlCommand("SELECT row FROM Data WHERE studentNo = '" &
secondStudentSV & "' AND population = '" & HFVPopulation & "'", con)
 Dim secondTableRow As String = getTableRow2.ExecuteScalar()
Dim getTableColumn2 As New SqlCommand("SELECT columnTable FROM Data WHERE studentNo
= '" & secondStudentSV & "' AND population = '" & HFVPopulation & "'", con)
Dim secondTableColumn As Integer = getTableColumn2.ExecuteScalar()
Dim mutation2 As New SqlCommand("Update Data SET totalSlot = '" & secondNoOfSlot & "', row =
'" & secondTableRow & "', columnTable = '" & secondTableColumn & "'WHERE studentNo = '" &
firstStudentSV & "' AND population = '" & HFVPopulation & "'", con)
mutation.ExecuteNonQuery()
```

Figure 4.12: Coding for Mutation

4.2 Result

The APEGA system is expected to have a high accuracy for the result due to its scheduling optimization problem is solving by using Genetic Algorithm technique. Besides, this system is expected have low percentage of errors occurring when generating the schedule. This is because the system will keep on looping to find the highest fitness value in order to solve the scheduling constraints and ensure it is no clashing occurs when matching the PSM evaluator.

The expected result will be an evaluator slot that assigned with a list of student where most of their project type is match with the evaluator's expertise and it had been optimized. The following is the example of evaluator slot in Microsoft Word document that generated by APEGA system.



Evaluator Slot

1: (1) ADZHAR BIN KAMALUDIN (2) Imran Edzereiq bin Kamarudin

CA09060
CD09026
CD09038
Cd09024
CD09057

2: (1) AWANIS BINTI ROMLI (2) ZALILI BINTI MUSA

CB09043
CA09060
CD09021
CD09052
CD09038

Figure 4.13: Evaluator Slot that Generated by APEGA System

The system was expected to run without errors and capable of generate the evaluator slot with no clashing and fulfill all the constraints. The final fitness value must be reached the maximum fitness value in order to achieve an optimization evaluator slot.

4.3 Testing

For the testing part, alpha testing will be conducted to test on the user interface. These two types of testing is performed with the participation of user to test for the user-friendliness of the interface as the system is being built.

In this stage, there are total of 12 lecturers and 30 students will are tested. The generation and time that use to produce the result will be recorded. The final fitness value will determine whether the expected result is achieved and fulfill with the constraints that had been set.

4.3.1 Testing Suite

The following show the parameter and testing suite that use to test for the APEGA system. The test suite will be test for 10 times.

Parameter	Test Suite
Lecturer	12
Student	30

Table 4.1: Testing Suite

4.3.2 Testing Environment

The testing environment will affect the result. The time for producing the timetable will increase if using the low specification computer to run the system. The table below is the testing environment for the APEGA system.

Item	Environment
Processor	Intel Core 2 Duo, 2.00GHz
RAM	3GB
Operating System	Windows Vista Home

Table 4.2: Testing Environment

4.3.3 Testing Result

The testing result is based on the time and generations needed to produce the evaluator slot. The expected result for final fitness value must be reached at least 50% which most of the student were assigned with the evaluators who are having the expertise that same with their project field. The testing result is showing in the table below.

Test Case	Mutation (%)	Fitness value (Max)	Final Fitness Value	Final Fitness Value (%)	Generation	Time (minutes)
1	5	60	31	51.6	10	2.40
2	5	60	33	55.0	10	2.46
3	5	60	29	48.3	10	2.75
4	5	60	36	60.0	10	2.48
5	5	60	31	51.6	10	2.46
6	5	60	39	65.0	10	2.44
7	5	60	37	61.6	10	3.01
8	5	60	28	46.7	10	2.39
9	5	60	43	71.7	10	3.11
10	5	60	38	63.3	10	2.43

Table 4.3: Testing Result

From the testing result, average 75% of the test cases have achieved more than 50% fitness value from the maximum fitness value. The time for testing one case which to

process about 12 evaluators and 30 students is take around 2 to 4 minutes. All of the test cases also reach the maximum generation which had been set to 10 by the system. This is because it is hard and very less chance for each case to achieve the maximum value as all the evaluator have different expertise field and the number for the student's project is different in each type.

As overall, the testing result could and produce the most optimum result for the assignation of PSM evaluator due to the uncertainty and unbalance between the numbers of student's project type and the evaluator's expertise. However, the result is still acceptable as it is success to solve at least 50% of the optimization problem of the PSM evaluator assignation compare to matching manually.

CHAPTER 5

CONCLUSION

This chapter is briefly discussed the conclusion of this project. This project is aimed to develop a system of Assignation of PSM Evaluator using Genetic Algorithm (APEGA). The overall performance of this research is acceptable.

5.1 Conclude on Chapter 1

Matching the optimum PSM evaluators for the students is a complex process due to it had a lot of complex constraints and optimization problems. Therefore, the objective of this project is to study on Genetic Algorithm which can be a technique that to use in solving optimization problem. Genetic Algorithm will be implemented into Assignation of PSM Evaluator application, thus the system can optimize the assignation of PSM evaluator for the student based on their project type and evaluator's expertise instead of the current manually matching.

5.2 Conclude on Chapter 2

Extensive research has been done and shows the exploitation of the Genetic Algorithms which have robust properties and demonstrating their capabilities across a broad range of optimization problems. This means that Genetic Algorithm capable to obtain the best and optimal solution for the problem by performing its operation. Thus, Genetic Algorithm is able to help in solving the main optimization problem of the presentation scheduling which is the evaluator can be distribute to evaluate the student's project which is based on his or her expertise field.

5.3 Conclude on Chapter 3

The development for the system of Assignation of PSM Evaluator using Genetic Algorithm (APEGA) is involved with customer. The customer's requirement and feedback is the main issues that use in the system development. The output of the project will be the most likely with the customer's needs. Therefore, a development model which involving with client was designed to develop this system. The development model consists of 9 phases which is starts from planning, followed by requirements gathering, analysis, design, build prototype, customer evolution, refine requirements, development and testing.

5.4 Conclude on Chapter 4

APEGA system is implement using Genetic Algorithm technique which starting by generate random population. The fitness value for each population will be calculated once the population was randomly generated. The population with highest fitness value will be chosen to pass to the next generation by Elitism Selection to remain its data that produce the highest fitness value. Crossover operation will only be performed on the evaluator slot that does not fulfill the constraints or having lowest fitness value to change over the assignation of students in order to produce new evaluator slot with higher fitness value. The mutation operation will be perform to change over the students between two selected evaluator slot and it will be done in low rate to avoid from slowing down the process of getting optimum solution.

The expected result of APEGA system is to have a high accuracy result due to its scheduling optimization problem is solving by using Genetic Algorithm technique. Besides, this system is expected have low percentage of errors occurring when generating the schedule. This is because the system will keep on looping to find the highest fitness value in order to solve the scheduling constraints and ensure it is no clashing occurs when generating the schedule.

5.5 Overall Conclusion

For the overall of conclusion, with the aid of APEGA system, it is able to minimize the other problems of optimization that had been found that during the assignation of PSM evaluator. Thus, this system is developed to assist the Faculty of Computer System and Software Engineering (FSKKP) for University Malaysia Pahang (UMP) in optimizing the assignation of PSM evaluator for the student based on their project field and evaluator's expertise. The APEGA system is able to developing a well-distributed matching between PSM evaluators and students by overcoming the relevant constraints in an intelligent way and help to reduced human energy. Therefore, the time consumption that uses to assign the optimum PSM evaluator for the student can be minimized in the same time by comparing with the current manually matching.

5.6 Constraint and Limitation

There are several constraints and limitations were identified through the APEGA system as following.

APEGA system is taken a longer time to produce the result due to the system have to run and evaluate through few populations to gain the optimal result. Therefore, this system only allow to produce a small scope of data and it haven achieve the best result.

Besides that, the system's user who is PSM coordinator has to group the evaluator by using the system one by one. APEGA system could not auto grouping the evaluator due to the number of evaluator in each department is different and uncertain. This will be one of the limitations for APEGA system.

5.7 Further Research

For the further research, the system can be enhancing to process a large number of data for the assignation of PSM evaluator due to the current prototype only can process a small scope of the data.

Besides that, the APEGA system is taken a longer processing time to produce the result. Therefore, the processing time for the system can be enhance by using other more high performing coding and also using high specification computer to run the system.

For the current research, Genetic Algorithm was found that is one of the suitable optimizer tools that use to solve the optimization problem such like scheduling. In further research, Genetic Algorithm technique will be researching for more detail and enhance its functionality in other application such as scheduling process.

REFERENCES

[1] S.F. Smith, *A Constraint-based Framework for Reactive Management of Factory Schedules*, Intelligent Manufacturing Proceedings, 1998.

[2] H.T. Amrine, J.A. Ritchey, C.L. Moodie, *Manufacturing, Organizing and Management*, Prentice Hall International Inc., USA, 1993.

[3] E.K. Burke, S. Petrovic, *Recent Research Directions in Autometed Timetabling*, European Journal of Operational Research 140 (2002) 266–280 , 2002.

[4] P.F Dutot, G. Mouni'e, D.Trystram, *Operating System Internals and Design Principles*, 5TH International Edition, Prentice Hall, 2004.

[5] S.A Kareem, A.Moitra, *Advances in Schedule Optimization with Genetic Algorithm*, GE Global Research, 2003.

[6] E. Burke, J. Kingston, K. Jackson, R. Weare, *Automated University Timetabling: The state of the art*, The Computer Journal 40 (9), 565–571, 1997.

[7] S.M. Ruegsegger, *Intelligent Scheduling Optimization Using A Rule-Based artificial Neural Network*, Case Western Reserve University, Cleveland, Ohio 44106.

[8] D. Borissove, *Optimal scheduling for Dependent Details Processing using MS Excel Solver*, Institute of Information Technologies, 2008.

[9] D. Beasley, D.R. Bull, R.R. Martin, *An Overview of Genetic Algorithms: Part 1,Fundamentals*, University Computing, Vol. 15, No.2, pp. 58-69, Department of Computing Mathematics, University of Cardiff, UK, 1993.

[10] T.V. Mathew, *Genetic Algorithm*, Department of Civil Engineering, Indian Institute of Technology Bombay, Mumbai-400076.

[11] D. Abramson, *Constructing School Timetables using Simulated Annealing: Sequential and Parallel Algorithms*, Management Science, Vol 37, pp.98 -113, 1991.

[12] D.Charles. *Charles Darwin: The Theory of Evolution. Truth & Reality*, 1997, Available on http://www.spaceandmotion.com/Charles-Darwin-Theory-Evolution.html, (November 14, 2011)

[13] U. Fissgus, *Scheduling using Genetic Algorithms*, Institut Fur Informatik, University Halle-Wittenberg, Germany.

[14] J.E. Yang, M.J. Hwang, T.Y. Sung, Y. Jin, *Application of Genetic Algorithm for Reliability Allocation in Nuclear Power Plants*, Korea Atomic Energy Research Institute, South Korea, Dec. 8th 1998.

[15] K.S. Tang, K.F. Man, S. Kwong, Q. He, *Genetic Algorithm and Its Applications*, IEEE Signal Processing Magazine, November 1996.

[16] M. Obitko, *Genetic Algorithms Description - Introduction of Genetic Algorithms. Genetic Algorithms*, 1998, Available on http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php, (November 18, 2011)

[18] N. Bouvanasilan, G.R. Ramani, V. Seenuvasan, *A perspective view on Travelling Salesman Problem using Genetic Algorithm*, Pondicherry Engineering College, Pondicheery University, India, 2009.

[19] Jankovic, Mladen. *Using Genetic Algorithm Library For Class Schedule. CoolSoft.* March 03, 2008. Available on http://www.coolsoftsd.com/ArticleText.aspx?id=1. (October 22, 2011)

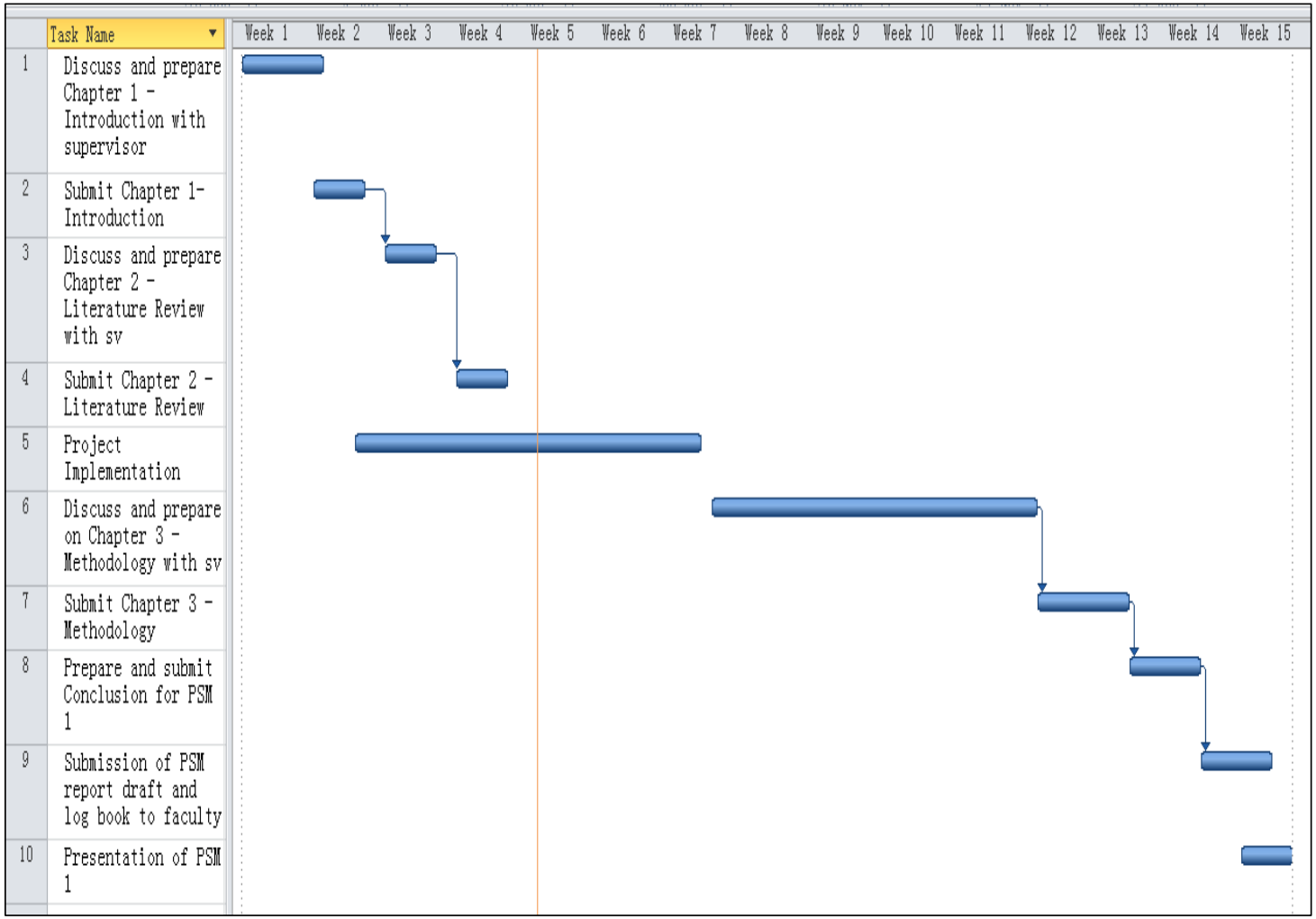
[20] B. Y. Liong, *Faculty Timetabling using Genetic Algorithm*, University Malaysia Pahang, Pahang, 2011.

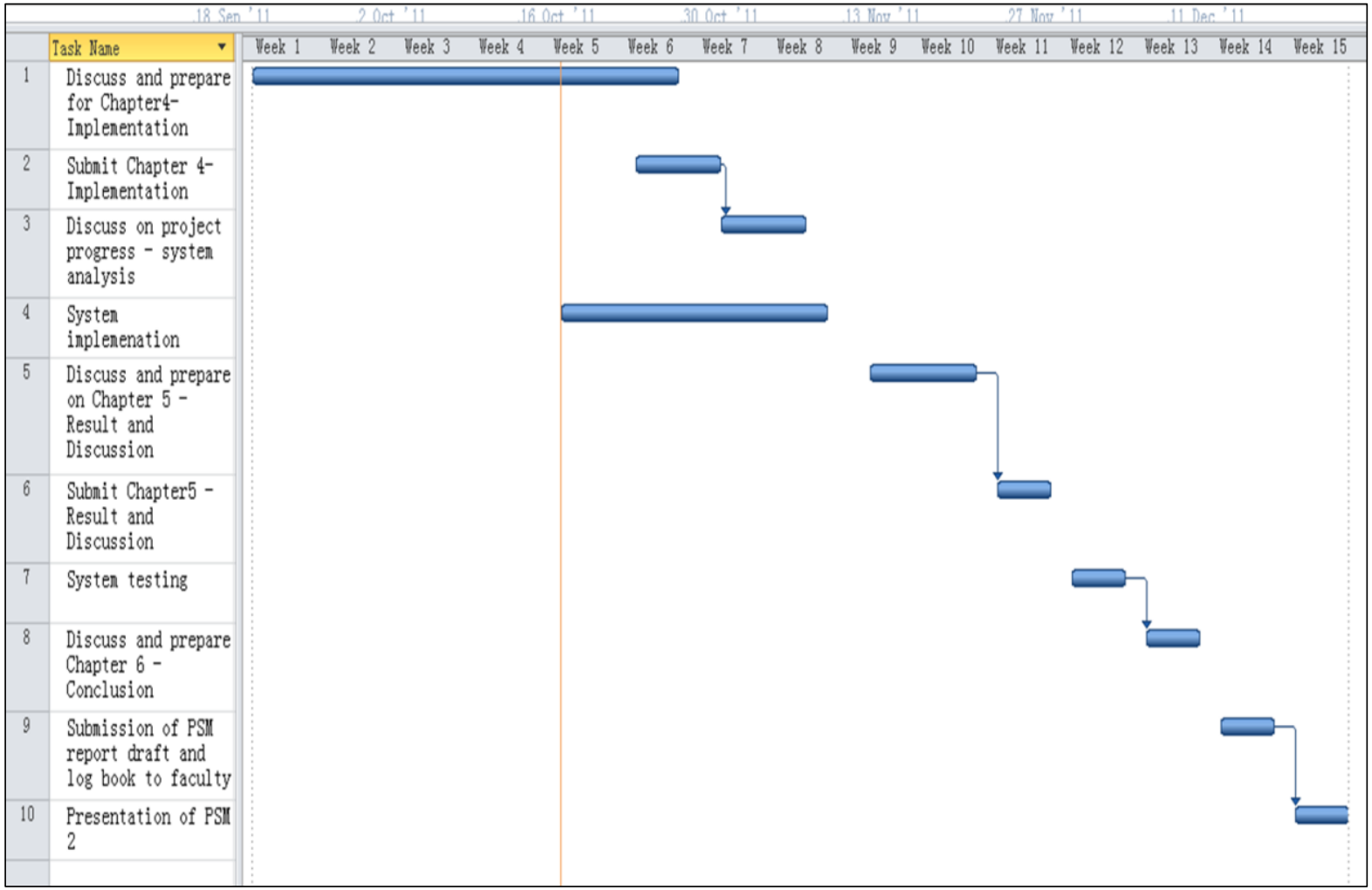
[21] Jankovic, Mladen. *Using Genetic Algorithm Library For Class Schedule. CoolSoft.* March 03, 2008. Available on http://www.coolsoftsd.com/ArticleText.aspx?id=1 (December 3, 2011)

APPENDICE

APPENDIX A:

GANTT CHART


Task Name
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
1 Discuss and prepare Chapter 1 - Introduction with supervisor
2 Submit Chapter 1- Introduction
3 Discuss and prepare Chapter 2 - Literature Review with sv
4 Submit Chapter 2 - Literature Review
5 Project Implementation
6 Discuss and prepare on Chapter 3 - Methodology with sv
7 Submit Chapter 3 - Methodology
8 Prepare and submit Conclusion for PSM 1
9 Submission of PSM report draft and log book to faculty
10 Presentation of PSM 1


18 Sep '11
2 Oct '11
16 Oct '11
30 Oct '11
13 Nov '11
27 Nov '11
11 Dec '11
Task Name
Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12
Week 13
Week 14
Week 15
1 Discuss and prepare for Chapter4- Implementation
2 Submit Chapter 4- Implementation
3 Discuss on project progress - system analysis
4 System implemenation
5 Discuss and prepare on Chapter 5 - Result and Discussion
6 Submit Chapter5 - Result and Discussion
7 System testing
8 Discuss and prepare Chapter 6 - Conclusion
9 Submission of PSM report draft and log book to faculty
10 Presentation of PSM 2

APPENDIX B:

DATABASE

Results | Messages

	row	totalSlot	population	studentID	studentNo	columnTable	groupFV	projectType	supervisor	fvLect1	fvLect2	fvSupervisor
1	2	1	3	CD09038	5	1	1	Software Engineering	TUTY ASMAWATY BINTI ABDUL KADIR	0	0	1
2	5	1	3	CA09060	1	1	1	Networking System and Security	Imran Edzereiq bin Kamarudin	0	0	1
3	3	2	3	CD09057	7	1	6	Graphic and Multimedia	RUZAINI BIN ABDULLAH ARSHAH	1	1	1
4	2	2	3	CD09052	6	1	6	Software Engineering	TUTY ASMAWATY BINTI ABDUL KADIR	1	0	0
5	1	1	3	CD09024	2	1	1	Computer Science	ZALILI BINTI MUSA	1	0	1
6	4	2	3	CD09026	3	1	6	Computer Science	ZALILI BINTI MUSA	0	0	1
7	2	1	3	CB09012	15	1	1	Software Engineering	ABDULLAH BIN EMBONG, PROFESOR. DR.	0	0	1
8	5	2	3	CB09043	10	1	6	Software Engineering	ABDULLAH BIN MAT SAFRI	1	0	1
9	5	3	3	CA09058	12	1	2	Networking System and Security	JASNI BINTI MOHAMAD ZAIN, PROFESO...	0	0	1
10	1	2	3	CD09021	9	1	6	Graphic and Multimedia	ADZHAR BIN KAMALUDIN	1	1	1
11	3	3	3	CD09027	4	1	2	Computer Science	ZALILI BINTI MUSA	0	0	1
12	4	3	3	CD09077	11	1	2	Graphic and Multimedia	ZALILI BINTI MUSA	1	0	1
13	1	3	3	CD09066	13	1	2	Software Engineering	CHE YAHAYA BIN YAACOB	0	0	1
14	2	3	3	CD09049	8	1	2	Graphic and Multimedia	RUZAINI BIN ABDULLAH ARSHAH	1	0	1
15	3	1	3	CA09016	14	1	1	Networking System and Security	Imran Edzereiq bin Kamarudin	0	0	1

Query executed successfully. | USER-PC (10.0 RTM) | User-PC\User (54) | master | 00:00:00 | 15 rows