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
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

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>i</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF APPENDICES</td>
<td></td>
<td>x</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td></td>
<td>xi</td>
</tr>
</tbody>
</table>

1 INTRODUCTION 1

1.1 Introduction 1
1.2 Problem Statement 2
1.3 Objectives 3
1.4 Scope 3
1.5 Thesis Organization 4

2 LITERATURE REVIEW 6

2.1 Scheduling 6
  2.1.1 Scheduling Problem 7
2.2 Genetic Algorithm 8
  2.2.1 Genetic Algorithm Procedures 8
  2.2.2 Genetic Algorithm Operations 10
2.2.2.1 Encoding 11
2.2.2.2 Selection 11
2.2.2.3 Crossover 12
2.2.2.4 Mutation 13

2.3 Existing Genetic Algorithm Application 14
2.3.1 Genetic Algorithm Class Schedule 14
2.3.2 Faculty Timetabling using Genetic Algorithm 19

2.4 Conclusion 23

3 METHODOLOGY 25

3.1 Development Model 25
3.1.1 Justification of Planning Development Model 26

3.2 Planning 28
3.2.1 Consultation with Supervisor 28
3.2.2 Choosing Project Title 28
3.2.3 Definition of Problem Statements 28
3.2.4 Definition of Research Objectives and Project Scopes 29
3.2.5 Planning for Project Milestone 29

3.3 Initial Requirements 29
3.3.1 Meeting with Client 29
3.3.2 Customer’s Requirements and Constraints 30
3.3.3 Specification Requirements 31
3.3.3.1 Software Specification 31
3.3.3.2 Hardware Specification 32

3.4 System Analysis 32
3.4.1 Analysis Based on Customer’s Requirements 33
3.4.2 Analysis on Implementation Technique 33

3.5 Design 33
3.5.1 System Design 33
3.5.1.1 Input Data of System 35
3.5.1.2 Output Data of System 35
3.5.1.3 Assumption of System 36
3.5.2 Software Design 36
  3.5.2.1 Encoding 36
  3.5.2.2 Fitness Evaluation 37
  3.5.2.3 Crossover 39
  3.5.2.4 Mutation 40
3.6 Prototype 40
3.7 Customer Evaluation 41
3.8 Refine Requirement and Update 41
3.9 Development and Implementation 42
3.10 Testing 42

4 IMPLEMENTATION & TESTING RESULT 43

4.1 Implementation 43
  4.1.1 Generate Random Population 43
  4.1.2 Fitness Value Calculation Function 46
  4.1.3 Elitism Selection 49
  4.1.4 Crossover 50
  4.1.5 Mutation 52
4.2 Result 54
4.3 Testing 55
  4.3.1 Testing Suite 55
  4.3.2 Testing Environment 55
  4.3.3 Testing Result 56

5 CONCLUSION 58

5.1 Conclude on Chapter 1 58
5.1 Conclude on Chapter 2 58
5.3 Conclude on Chapter 3 59
5.4 Conclude on Chapter 4 59
5.5 Overall Conclusion 60
5.6 Constraints and Limitation 60
5.7 Further Research 61

REFERENCES 62

APPENDICES 64
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1: Software Specification for APEGA</td>
<td>31</td>
</tr>
<tr>
<td>Table 3.2: Hardware Specification for APEGA</td>
<td>32</td>
</tr>
<tr>
<td>Table 4.1: Testing Suite</td>
<td>55</td>
</tr>
<tr>
<td>Table 4.2: Testing Environment</td>
<td>55</td>
</tr>
<tr>
<td>Table 4.3: Testing Result</td>
<td>56</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 2.1: GA Cycle</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.2: The Basic of Genetic Algorithm Operations</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.3: Example of Encoding for Two Chromosomes</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.4: A Roulette-Wheel that Represents of Five Individual and</td>
<td>12</td>
</tr>
<tr>
<td>Their Fitness Values</td>
<td></td>
</tr>
<tr>
<td>Figure 2.5.1: One Site Crossover Operation</td>
<td>12</td>
</tr>
<tr>
<td>Figure 2.5.2: Two Site Crossover Operation</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.6: Mutation generation</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.7: Chromosome Representation</td>
<td>16</td>
</tr>
<tr>
<td>Figure 2.8: Crossover of Genetic Algorithm Scheduler</td>
<td>17</td>
</tr>
<tr>
<td>Figure 2.9: Interface of Genetic Algorithm Schedule with its Fitness</td>
<td>19</td>
</tr>
<tr>
<td>Their Fitness Values</td>
<td></td>
</tr>
<tr>
<td>Figure 2.10: List of Hard Constraints</td>
<td>20</td>
</tr>
<tr>
<td>Figure 2.11: The Relationship between Chromosomes and Genetics for</td>
<td>21</td>
</tr>
<tr>
<td>FTGA</td>
<td></td>
</tr>
<tr>
<td>Figure 2.12: Elitist Selection of FTGA</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.13: Illustration of Crossover Process for FTGA</td>
<td>22</td>
</tr>
<tr>
<td>Figure 2.14: Illustration of Mutation Process for FTGA</td>
<td>23</td>
</tr>
<tr>
<td>Figure 3.1: Flow of Model</td>
<td>26</td>
</tr>
<tr>
<td>Figure 3.2: Flow Chart of PPSGA System</td>
<td>34</td>
</tr>
<tr>
<td>Figure 3.3: Example of Evaluator Slot</td>
<td>36</td>
</tr>
<tr>
<td>Figure 3.4: Relationships between Elements of Genetic Algorithm</td>
<td>37</td>
</tr>
<tr>
<td>Figure 3.5: Evaluation of Fitness Values</td>
<td>38</td>
</tr>
<tr>
<td>Figure 3.6: Illustration of Crossover</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.7: Illustration of Mutation</td>
<td>40</td>
</tr>
</tbody>
</table>
Figure 3.8: Prototype of A Set of Evaluator Slot that Generated by APEGA

Figure 4.1: Coding for Check Evaluator Slot Availability
Figure 4.2: Coding for Generate Random Population
Figure 4.3: Coding for Retrieve Data from StudentInfo Table
Figure 4.4: Coding for Retrieve Current Project’s type of the Student
Figure 4.5: Coding for Retrieve Evaluator’s expertise From Group Evaluator Table
Figure 4.6: Coding for Fitness Value Calculation between Student Project Type and Evaluator’s expertise
Figure 4.7: Coding for Elitism Selection
Figure 4.8: Coding for Assign Highest Fitness Value
Figure 4.9: Coding for Retrieve Data that Have Low Fitness Value
Figure 4.10: Coding for Crossover
Figure 4.11: Coding for Set Mutation Rate
Figure 4.12: Coding for Mutation
Figure 4.13: Evaluator Slot that Generated by APEGA System
## LIST OF APPENDICES

<table>
<thead>
<tr>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPENDIX A: Gantt Chart</td>
<td>65</td>
</tr>
<tr>
<td>APPENDIX B: Database</td>
<td>68</td>
</tr>
</tbody>
</table>
**LIST OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>APEGA</td>
<td>Assignation of PSM Evaluator using Genetic Algorithm</td>
</tr>
<tr>
<td>PSM</td>
<td>Projek Saujana Muda</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
</tbody>
</table>
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 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.

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.

![GA Cycle](image)

**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 bigger 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](image)

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.

![Figure 2.3: Example of Encoding for Two Chromosomes](image)

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.
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.