SOLVING AN APPLICATION OF UNIVERSITY COURSE TIMETABLING PROBLEM BY USING GENETIC ALGORITHM

NORHANA BINTI SHAIBATUL KHADRI

MASTER OF SCIENCE (DECISION SCIENCE)
UNIVERSITI UTARA MALAYSIA
2022

SOLVING AN APPLICATION OF UNIVERSITY COURSE TIMETABLING PROBLEM BY USING GENETIC ALGORITHM

A thesis submitted to UUM College of Arts and Sciences
In partial fulfilment of the requirements for the degree of
Master of Science (Decision Science)
School of Quantitative Sciences (SQS)

By

NORHANA BINTI SHAIBATUL KHADRI

PUSAT PENGAJIAN SISWAZAH
(CENTER FOR GRADUATE STUDIES)
UNIVERSITI UTARA MALAYSIA

PERAKUAN KERJA/DISERTASI
(Certification of Dissertation Work)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

NORHANA BINTI SHAIBATUL KHADRI

calon untuk Ijazah
(candidate for the degree of) **MASTER of SCIENCE (DECISION SCIENCE)**

telah mengemukakan tesis/disertasinya yang bertajuk
(has presented his/her project paper of the following title)

SOLVING AN APPLICATION OF UNIVERSITY COURSE TIMETABLING PROBLEM BY USING GENETIC ALGORITHM

seperti yang tercatat di muka surat tajuk dan kulit tesis/disertasi
(as it appears on the title page and front cover of project paper)

bahasa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan, dan liputan bidang ilmu yang memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : **17 MEI 2022**
(That the thesis/dissertation is acceptable in form and content, and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate through an oral examination held on ***17 May2022***

Pengerusi Viva *(Chairman for Viva)*	: Puan Zakiah binti Hashim	Tandatangan : *(Signature)*
Penilai Dalam 1 *(Internal Assessor 1)*	: Dr. Aida Mauziah binti Benjamin	Tandatangan : *(Signature)*
Penilai Dalam 2 *(Internal Assessor 2)*	: Prof. Madya Dr. Ruzelan bin Khalid	Tandatangan : *(Signature)*
Penyelia Utama *(Principal Supervisor)*	: Dr. Rosshairy binti Abdul Rahman	Tandatangan : *(Signature)*
Penyelia Kedua *(Co-Supervisor)*	: Prof. Madya Dr. Syariza binti Abdul Rahman	Tandatangan : *(Signature)*
Dekan Pusat Pengajian Sains Kuantitatif *(Dean, School of Quantitative Sciences)*	: Prof. Madya Dr. Mohd Kamal bin Mohd Nawawi	Tandatangan : *(Signature)*

Tarikh *(Date)* : 12 September 2022

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the University Library may take it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Request for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman

Abstract

Generating timetables for academic institutions is a complex problem. This is due to many constraints involved whether they are vital or desirable, which are known as hard and soft constraints. The problem becomes more complicated and difficult to solve as the number of courses increase. Moreover, generating manual timetables is challenging and time-consuming, particularly to meet lecturers' preferences. Thus, it is crucial to establish an automated course timetable system. Many efforts have been made using various computational heuristic methods to acquire the best solutions. Among the approaches, genetic algorithm (GA), constructed based on Darwin's theory of evolution, becomes the renowned approach to solve various types of timetabling problems. Therefore, this study produces the best timetable using GA to solve clashed courses, optimize room utilization and maximize lecturers' preferences. Data of 41 course sections from 17 courses offered in semester A172 were taken from Decision Science Department, School of Quantitative Sciences (SQS). The phases in GA involves a number of main operators which are population initialization, crossover and mutation. The best parameter setting for GA was determined through combination of different mutation rate, population and iteration. The simulation results of GA show that this method is able to produce the best fitness value that satisfied all hard and soft constraints. There are no clashes either between lecturers or lecture rooms, and lecturers' preferences were satisfied. The system can help SQS or any other academic schools or institutions to easily develop course timetabling in the coming semesters.

Keywords: Course timetabling problem, Lecturers' preferences, Genetic algorithm, Population-based metaheuristic

Abstrak

Penjanaan jadual waktu kursus untuk institusi akademik merupakan masalah yang kompleks. Ini disebabkan oleh banyak kekangan yang terlibat sama ada ia penting atau wajar, yang dikenali sebagai kekangan keras dan lembut. Masalah menjadi lebih rumit dan sukar untuk diselesaikan apabila jumlah kursus meningkat. Tambahan pula, penjanaan jadual waktu secara manual adalah mencabar dan memakan masa, terutamanya untuk memenuhi keinginan pensyarah. Maka, adalah penting untuk membangunkan satu sistem jadual kursus automatik. Pelbagai usaha telah dilakukan menggunakan pelbagai kaedah pengiraan heuristik untuk memperoleh penyelesaian terbaik. Di antara pendekatan tersebut, algoritma genetik (GA), yang dibina berdasarkan teori evolusi Darwin, menjadi pendekatan yang terkenal untuk menyelesaikan pelbagai jenis masalah jadual waktu. Oleh itu, kajian ini menghasilkan jadual waktu terbaik menggunakan GA untuk menyelesaikan pertindihan kursus, mengoptimumkan penggunaan bilik dan memaksimumkan keinginan pensyarah. Data 41 seksyen kursus daripada 17 kursus yang ditawarkan pada semester A172 diambil dari Jabatan Sains Pemutusan, Pusat Pengajian Sains Kuantitatif (SQS). Fasa-fasa GA melibatkan beberapa operator utama iaitu pemulaan populasi, persilangan dan mutasi. Tetapan parameter terbaik untuk GA ditentukan melalui gabungan kadar mutasi, populasi dan lelaran yang berbeza. Hasil simulasi GA menunjukkan bahawa kaedah ini mampu menghasilkan nilai kecergasan terbaik yang memenuhi semua kekangan keras dan lembut. Tiada pertindihan sama ada di antara pensyarah atau bilik kuliah, dan keinginan pensyarah dipenuhi. Sistem ini boleh membantu SQS atau mana-mana pusat pengajian atau institusi lain dalam membangunkan jadual waktu kursus untuk kuliah pada semester yang akan datang dengan mudah.

Kata kunci: Masalah jadual waktu kursus, Keinginan pensyarah, Algoritma genetik, Metaheuristik berasaskan populasi

Acknowledgement

First and foremost, all praise to Allah for His blessing and guidance, I able to complete this research successfully.

I would like to express my utmost gratitude to many people who have helped in finishing this research both directly and indirectly. Millions of thanks go to my supervisors, Dr. Rosshairy Abdul Rahman and Assoc. Prof. Dr. Syariza Abdul Rahman for contributing ideas and thoughts this research, also their supervision and support to help the progression and smoothness throughout the process.

I would also like to sincerely thank all my supportive friends and all the management and staff from the Deputy Vice Chancellor (Research & Innovation), AACSB Accreditation Unit, Research and Innovation Management Centre (RIMC), Centre for Testing, Measurement & Appraisal (CeTMA) and School of Quantitative Science (SQS). Special thanks to my husband (Muhammad Nazli bin Ismail), beloved family and friends, for the encouragement and support. Without their support, this master's journey would not have been fulfilled.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Background of Study

Resource allocation exists in a wide variety of domain including healthcare institution, transportation, sport, industrial environments, and education. It is the matter how the resources are being allocated to perform a collection of tasks over time. In education domain such as school and university, resource allocation is always being referred as timetabling (Wong, Goh & Likoh, 2022; Petrovic & Burke, 2004).

Diaz-Parra et al. (2022) and Wren (1996) defined timetabling as the allocation of resources to objects placed in space time, depending on constraints, to fulfil a set of desirable objectives the nearest likely. Nuntasen and Innet (2017) stated that university timetabling problem is the arrangement to fulfil compliance and relation of courses, lecturers, classrooms, students, day, and time.

A general university timetabling can comprise of sub-problems such as lecturer assignment, course timetabling, student timetabling and classroom assignment (Gunawan, Ng & Poh, 2012; Bashab et al., 2020). According to Tan et. al. (2021), and Adewumi, Sawyer and Montaz Ali (2009), these problems are categorized under NP-hard problem that concerns with the allocation of certain resources, based on constraints with the goal of achieving a set of stated objectives to the best possible level, which is very difficult to solve exactly or optimally.

Assignment process in university timetable entails in allocating lecturers to courses or course sections and allocating courses to classrooms. Many studies only focus on one sub-problems, though actual problems of university timetabling always comprise a grouping of some sub-problems.

For example, works of Ngo et al. (2021), Schniederjans and Kim (1987), Wang (2002), Gunawan, Ng and Poh (2008), and Hosny (2012) concentrated on resolving the lecturer assignment problem without taking into consideration how the courses being allocated to time slots. While Wang (2002) focused on the lecturer assignment problem in which almost every educational institution needs to handle regularly. Alves et al. (2022) and Lee et al. (1997) defined lecturer assignment as an assignment used to decide which lecturer will teach what courses by considering multiple constraints.

In another study which focused on university classroom problem, the classroom is arranged by considering the distance of students' movement between courses (Wang, Jeng & Zhang, 2019). While Mokhtari et al. (2021) mentioned courses need to be allocated to particular time slots in working days by considering ample space of classroom and the number of students registered.

There are two categories of constraints, which are hard and soft constraints (Ghaffar et. Al., 2022; Burke & Petrovic, 2002). Hard constraints could not be disregarded completely as to ensure the solution is free from conflict, while the soft constraints which relate to objective function should be maximally satisfied in the final allocation. A lot of institutions still perform this task in a conventional way, which consumes time and hard work (Gunawan, Ng & Poh, 2007; Muklason et al., 2018). Moreover, the timetables generated manually do not usually meet the preferences of all lecturers.

A timetable is effective when it is feasible and brings certain quality features that keep users' satisfaction at a certain grade. Thus, computing the tasks of scheduling and planning can reduce the responsibilities of an administrator. In addition, it will help to improve the productivity, quality of education, services, and life (Nuntasen & Innet, 2007; Muklason et al., 2018).

1.2 Issues in University Course Timetabling Problem

The scheduling of courses at many universities is still done manually and this allows for human error to occur, causing clashes between courses. Issues related to hard constraints concern physically unbearable. For instance, the resources like lecturers or students assigned in two different classroom at one time. Alike, same group of students are not allowed to be taught by different lecturers for two separate courses at one time slot.

Also important to note that, lecturers or students are not authorized to be in more than one classroom at an accustomed time. Additional, other essential resources for instance the lecturers and classrooms must be feasible for respectively time slot.

If the schedule is not being generated in a proper way, then it might come into overlapping the problem (clashes) and contribute to unsatisfied lecturers. Thus, it is necessary to arrange an automatic course schedule with an optimization method approach so that it can facilitate complex scheduling and can cover the many constraints in scheduling.

1.3 Approaches to University Course Timetabling Problem

Many contributions and approaches in the previous works have been proposed to solve the timetabling problems. Solutions used to solve this problem can be classified into categories as follows: (1) exact methods including Goal Programming (Lindahl, Serensen & Stidsen, 2018; Badri, 1991) and Integer/Linear programming (IP/LP) method (Maldonado-Matute, Gonzalez Calle & Celi Costa, 2020; Daskalaki & Birbas, 2005) (2) Heuristic and metaheuristic methods including hill climbing (Yusoff & Roslan, 2019; Hosny, 2012), Genetic Algorithm (GA) (Wang, 2002; Qi & Xu, 2012; Turki, 2014; Wang, Geng & Zhang, 2019; Rezaeianah, Matoori & Ahmadi, 2021), Simulated Annealing (SA) (AlHadid et al., 2020), and Tabu Search Algorithm (TS) (Chen et al., 2020) and (3) intelligent novel methods such as hybrid approaches (Bashab et al. 2020; Gunawan, Ng & Poh 2008).

Based on reviews done by Babaei, Karimpour, and Hadidi (2015) and work supported by Alghamdi et al (2020), exact methods have been proven to be unsuitable or incompetent in solving university course timetabling problem, but rather they have simpler operation. This is due to the complexity of exact methods will rise as number of resources rises; thus, analysis of this type of problem categorized as NP-hard problems with exponential time complexity problem. However, exact methods may be applied as a solution method for university course timetabling problem if the complexity of time and space is not important. In the meanwhile, application of metaheuristic methods and novel intelligent methods can shorten much time if compared to exact methods.

Metaheuristic methods start with initial solutions and apply search strategies which attempt to prevent from getting stuck in local optima. These search algorithms can produce high quality solutions. According to Chen and Zheng (2020) metaheuristic could generate better solutions than those being generated from sequential heuristics alone. However, metaheuristic usually need the adjustment of parameters and have considerable computational cost. Another disadvantage of metaheuristic is one solution that is used on one problem does not usually applicable to other problems. Thus, it is difficult to state bounds on solution quality. In assignment problem, GAs used to produce good quality solutions (Wang, 2002; Gunawan, 2008; Assi, Halawi & Haraty, 2018; Rezaeipanah, Abshirini & Zade, 2019). These studies become the motivation to further explore the GA and lecturer assignment problem.

1.4 Problem Statement

The task of allocating courses to the lecturers in UUM, specifically in Department of Decision Science, School of Quantitative Sciences (SQS) was being done manually by the undergraduate and postgraduate programmes' coordinator. Previously, lecturers were asked to fill in a course selection form where the task has become more complicated as the number of lecturers increases, as well as their preference on teaching day. Another consideration is the classroom size based on the number of students enrolled. Therefore, the implementation of automated university timetabling is very important in reducing the preparation time due to complexity in fulfil all the requirement constraints.

University timetabling problem associated with allocating the lecturers to the courses, time slots and classrooms by considering some requirements that need to be fulfilled. Timetables should comprise of assigned available courses to lecture classrooms of varying sizes. Previous studies have identified different constraints to optimize the objective function of university timetabling problem. For instance, different courses cannot be delivered in two different places at the same time by the same lecturer (Modibbo et.al, 2019). Perhaps conflicts between several soft constraints and as a consequence an adjustment is necessity among them. As an illustration, soft constraint for a course with 10 students enrolled will allocate an appropriate classroom that can fit the students' number. Conversely, perhaps some lecturers prefer classroom that can fit to only 30 students. The algorithm developed will expectantly assign a favoured configuration as preference of lecturer is reflected as soft constraints. The complexity of this problem motivates this study to be conducted by identifying the constraints that can optimize the lecturer's preferences while a number of constraints are adhered with.

Past research conducted by Wang (2002) applied GA to allocate 20 lecturers to 90 courses. The findings proved that GA may reduce the time used significantly on assigning lecturer to courses. Gunawan, Ng and Ong (2008) also focusing on assigning the lecturers to the course sections as to balance the lecturers' load by using GA. Comparison in terms of computational result show that GA produce better solutions than manual allocation done by administrator. However, both researches conducted have used different parameters setting. Therefore, this study will investigate the best parameter setting of GA as to generate university timetabling problem in Decision Science Department that strictly satisfying hard constraints and fulfilling the soft constraints as optimal as possible.

1.5 Research Question

i. What requirements must the university timetabling problem satisfy?

ii. How to design a GA to solve the university timetabling problem?

iii. What is the best parameter setting for GA in solving university timetabling problem?

1.6 Research Objective

In general, the aim of this study is to build a model using GA approach in solving university timetabling problem by optimizing lecturer's preferences based on preferred day, time slots and classroom size.

Specifically, the study focuses on:

i. To identify the requirements of university timetabling problem that needs to be satisfied in generating feasible timetable.

ii. To develop a GA model for solving university timetabling problem that satisfying constraints.

iii. To experiment the best parameter of GA model in solving University timetabling problem.

1.7 Significance of the study

The study attempts to offer valuable contribution towards the management of the university timetabling problem. It is a hope that this study able to demonstrate the practicability of developed algorithm as an effective tool for obtaining optimal solutions in general university course timetable. This study may also serve as a reference for future research on the similar topic of university timetabling problem where preferences on day and time, also assigning classroom based on number of students to be considered.

1.8 Scope of the Study

The study focuses on the university timetabling problem for Semester A172 at the Decision Science Department under School of Quantitative Sciences, Universiti Utara Malaysia considering courses provided for the undergraduate and postgraduate programs.

1.9 Summary

This thesis comprises of five chapters. Chapter 1 presents the essential background and motivation for the issues also scope and the aims of the study. Chapter 2 presents several timetabling problems and reviews the previous published studies related to university timetabling. Next, Chapter 3 explains the design of research methodology to achieve the objectives as described in this chapter. The first step in the methodology is problem definition/ boundary selection. This is then followed by data collection and mathematical formulation, implementation of GA approach, model validation and verification. Further, Chapter 4 presents the findings and discussion of this study. Findings include the output obtained from MATLAB. Lastly, Chapter 5 summarizes the overall research including the findings, research contributions, research limitations and recommendation for future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discussed some related articles on the application of university timetabling. This chapter begins by introducing a detail description about the classification of university timetabling problem, complexity, constraints and three classifications of algorithms being studied by past researchers is presented. Also, a brief overview of GA approach in solving courses timetabling problems is discussed.

2.2 The Classification of University Timetabling Problem

According to Burke and Petrovic (2002) supported by Tan et. al. (2021), educational timetabling problem consists of course and examination timetabling problems. Multi-dimensional assignment problem can be regard to the course timetabling problem where lecturers or students being allocated to courses and course sections. It also involves assigning courses to classrooms and timeslots within a week. This study focused on course timetabling problem and classroom assignment problem.

The actual timetabling problems at all times comprise a grouping of the sub-problems as shown in Figure 2.1, even if not entirely sub-problems might be applicable to a specific condition (Gunawan, Ng & Poh, 2012).

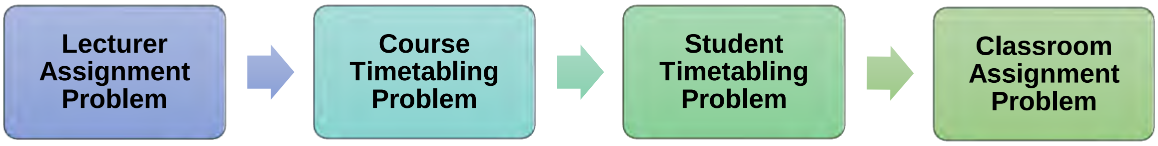


Figure 2.1 Types of university course timetabling problem

The next subsection will discuss further the definition and explanation of each classification of university course timetabling problem.

2.2.1 Lecturer Assignment Problem

The aim of lecturer assignment problem is to allocate lecturers to the courses in view of lecturers' preferences. According to Andrew and Collins (1971) and Mallicka (2021), the problem is an assurance that all courses will be staffed, and no lecturer is burdened which contributing to high effectiveness and preference ratings in lecturer assignments.

Some of the aspects to be considered in allocating the lecturers to the courses are lecturers' abilities and courses offered as to balance their workloads where the load denotes the assigned number of credits to each lecturer. Gunawan and Ng (2011) and Kusuma and Adiputra (2021) proposed SA and TS to solve the problem and the prearranged assignments might decrease the assignment problem scope and difficulty. Lecturer assignment problem have been a very challenging task by considering a range of criteria such as lecturer's qualifications, preferences, and courses requirements (Turki et al., 2014; Esteban, Zafra & Romero, 2018). Badri (1996), and Genc and O'Sullivan (2020) proposed a two-phase optimization process, which aimed to satisfactorily assign lecturers to their preferred courses and satisfy the lecturers' preferences in terms of time period for each courses to be taught. Wang (2002) applied GA and has considered multiple constraints such as accomplishment of teaching requirement, reasonable range of lecturer's overtime hours and academic qualification for solving lecturer assignment problem at Far East College, Taiwan.

2.2.2 Course Timetabling Problem

The purpose of problem in course timetabling is regard to the allocation of courses or course sections to time slots available. Course timetabling can be considered as the most popular problem being studied (Abramson, Amoorthy & Dang, 1999; Akkan, & Gulcu, 2018). This problem can be joint with another sub-problem. Though, it is frequently expected that the course timetabling problem is solved after solving the lecturer assignment problem (Al-Yakoob & Sherali, 2006; Arratia-Martinez, Avila-Torres, & Trujillo-Reyes, 2021). While Gunawan, Ng and Poh (2007), and Rezaeipanah, Matoori, and Ahmadi (2021) proposed hybrid algorithm to solve both lecturer assignment and course timetabling problem simultaneously in their study.

2.2.3 Student Timetabling Problem

The focal determination in this type of timetabling is the allocation of students to the course section while balancing section sizes and considering classroom sizes. As being stated by Carter and Laporte (1997), this problem arises especially when the courses have many sections.

2.2.4 Classroom Assignment Problem

Courses must be allocated to specific classrooms and time slots. To simplify, the courses are allocated to time slots, then followed by assigning courses to the classrooms (Carter & Laporte, 1997; Niknamian, 2019). Another consideration in this type of assignment is the location and size of the classroom (Botangen & Khan, 2014; Wang, Geng, & Zhang, 2021). Alves et al. (2022) mentioned that problems of classroom assignment may be regarded as straightforward compare to other types of sub-problems.

2.3 Complexity

A problem is NP-hard if solving it in polynomial time would make it possible to solve all problems in class NP in polynomial time. Alghamdi et al (2020) in his study, mention that university timetabling problem can be classified as NP-complete as it cannot be resolved in a sensible amount of time.

Scheduling problems belong to the class of NP-hard problems for which there is no polynomial time algorithm that exists to solve the problem to optimality (Iwankowicz & Taraska, 2018; Adewumi, Sawyer & Montaz; 2009). This problem comprise various shapes and forms (Al-Negheimish et al., 2018; Hosny, 2012) that involves the assignment of certain resources, with the purpose to fulfil a set of constraints to the uppermost level. The existence of constraints increase the problem difficulties.

2.4 Constraints of the University Timetabling Problem

Table 2.1 denotes a list of several common hard and soft constraints for the university timetabling problem presented in the literature.

Table 2.1 List of common constraints for University Timetabling Problem

Hard constraints	Author (s)
1. No lecturer who has been set in different room at the same time	Gozali and Fujimura (2020)
2. Any event should be scheduled in a suitable classroom size	Gozali and Fujimura (2020); Modibbo et.al (2019)
3. No conflict of classes	Wang, Geng and Zhang (2019)
Soft Constraints	**Author (s)**
1. Preferences of lecturers should be evaluated.	Badri (1996); Hosny (2012); Turki et.al (2014); Gunawan, Ng and Poh (2008); Babaei, Karimpour and Hadidi (2019)
2. Some lecturers should be scheduled in their preferred time	Gozali and Fujimura (2020)
3. Choose the classroom with smaller size	Wang, Geng and Zhang (2019)

2.5 Algorithms for University Course Timetabling Problem

There are numbers of approaches or methods to university course timetabling problems that have being studied by previous researchers. According to Babaei et. al (2019), methods employed to solve the university course timetabling problem can be categorized into three as follows: (1) Exact approach (2) Metaheuristic approaches and (3) intelligent novel approaches.

2.5.1 Exact Approach

Exact approaches can reach an optimal solution and to prove its optimality for every instance of a combinatorial optimization problem. However, exact approaches are unpractical to solve huge problems as the problems are very time-consuming (Dorneles et al., 2017; Saviniec et al., 2020). Often, only moderately sized or small cases can be solved in practical to unarguable optimality since the run time increases intensely with the case size. Examples of exact based techniques for solving university timetabling problem are Goal Programming (Gupta & Sinha, 2020; Badri, 1996) and Mixed Integer Programming (Qu et.al, 2017).

Gupta and Sinha (2020) and Badri (1996) demonstrated Goal Programming model's capability in solving assignment of lecturers to courses and next, to allocate the combinations to preferred time slot. The application proved to satisfy the department rules and policies, regarding the courses offered, while considering lecturers' preferences in teaching specific courses and time slot preferences. One of the earliest studies of lecturer assignment to course using Goal Programming proposed by Schniederjans and Kim (1987) has overcome the limitation of the previous model implemented by Harwood and Lawless.

Qu et.al (2017) developed Mixed-integer Linear Programming models to assign the most suitable teaching assistants to the tutorials class in a department by considering the capabilities differences and satisfying maximum workload. While, Rjoub (2020) studied courses timetabling based on hill climbing algorithm.

2.5.2 Metaheuristics

Metaheuristics approach is widely used in solving university timetabling and can be classified into population based and single solution-based approaches. Population based approaches includes GA (Zade, 2019; Alsmadi et al., 2011), Ant Colony Optimization (Mayer et.al, 2008; Xu & Zhou, 2020) and Partial Swarm Optimization (Abayomi-Alli et al., 2019), while single solution approaches includes TS (Chen et al., 2020), Variable Neighbourhood Search and Great Deluge Algorithm (Muklason, Irianti & Marom, 2019), and SA (Aycan & Ayav, 2008).

Saviniec et al. (2020) and Costa (1994) said that it is essential to build a metaheuristics in seeking a feasible solution. It can replace exact method which happen to be problematic to seek optimal solution for larger size.

Two phases are involved in constructing sequential assignments in heuristic approach, where in phase one, the process involves initiating solution that is feasible, and in phase two is to make improvement of the solution obtained in phase one in order to achieve the best optimal solution. The practice of metaheuristics commonly encounters the requirements of decision makers to generate satisfactory solutions in efficient way.

According to Mostafaie, Khiyabani, and Navimipour (2020, a metaheuristic leads and amends other heuristics to generate solutions that are not normally generated for local optimality. It is defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information to find efficiently near-optimal solutions (Koksal et al., 2021; Osman & Laporte, 1996).

Several applications of metaheuristics approach focusing specifically in solving lecturer assignment problem are concise in Table 2.2.

Table 2.2 Approaches in solving university timetabling problem

Authors	**Approach**
Wang (2002); Gunawan, Ng and Ong (2008); Qi and Xu (2012); DavidWilson, Davies, and Stanton (2013); Modibbo et.al (2019); Gozali and Fujimura (2020); Puspitasari and Moengin (2020)	Genetic Algorithm
Chen et al. (2020); Muklason, Irianti and Marom (2019)	Tabu Search
Goh, Kendall and Sabar (2019); Gunawan, Ng and Poh (2007); Gunawan, Ng and Poh (2008)	Simulated Annealing
Abayomi-Alli et al. (2019); Hossain et al. (2019)	Particle Swarm Optimization

Modibbo et.al (2019), Gozali and Fujimura (2020), Puspitasari and Moengin (2020), Berisha, Bytyci and Tershnjaku (2017), Nuntasen and Innet (2007) and Burke, Elliman and Weare (1994) proposed the GA as an efficient method to solve university timetabling problem. According to Sastry and Goldberg (2014), GA is a search method based on natural evolution and genetics. It can get globally optimal solutions even in the most complex of search spaces. The GA been used by Wang (2002), Gunawan, Ng and Ong (2008), Qi and Xu (2012), DavidWilson, Davies, and Stanton (2013) and Turki (2014), Modibbo et.al (2019), Gozali and Fujimura (2020) and Puspitasari and Moengin (2020) where they focused in solving university timetabling problem. The results proved that the proposed algorithm able to maximize lecturers' satisfaction subject to different constraints including department's requirement especially lecturers' preferences.

2.5.3 Intelligent novel methods

Past studies in solving university timetabling problem used intelligent novel methods such as hybrid approach (Gunawan, Poh & Ng, 2007), fuzzy theory-based approaches (Asmuni, Burke & Garibaldi., 2005) and Clustering Algorithm based approaches (Shatnawi et al., 2010). Abdullah and Hamdan (2008) proposed a combination of beneficial and competent features of other approaches in solving the timetabling problem. As individual approaches have some imperfections, thus solution quality from this combination can be improved. Besides, by reducing the imperfections of individual methods, the computation can be accelerated and contribute to a superior hybrid approach.

In solving a combination of lecturer assignment and course timetabling problem, Gunawan, Poh and Ng (2007) suggested hybrid algorithm that associates a greedy heuristic, modified Simulated Annealing and an Integer Programming. The results yield better solution as the number of variables and constraints increase by reducing the weaknesses of an Integer Programming approach.

In other studies of hybrid algorithm by Gunawan, Ng and Poh (2008), combination of two metaheuristics; SA and TS are proposed to solve lecturer assignment and scheduling problem simultaneously. The results of hybrid approach yield better solution compared to individual approach as hybrid exploit and combine the advantages of SA and TS.

2.6 Application of Genetic Algorithm in Lecturer Assignment Problem

The university timetabling problem obtained the attention of many researchers in many works, due to its importance as a practical application in many universities. This problem is considered to be complex to find a feasible solution because of the hard and soft constraints. According to Hosny and Fatima (2011), and Lindahl, Sørensen and Stidsen (2018), among the popular metaheuristics being applied to solve this problem is GA, which is an intelligent optimization technique that is fit for solving hard and highly constrained problems.

GA is a great algorithm to find optimized solutions, which model the principles of evolution; hence, is employed to solve university timetabling problems (Nuntasen & Innet, 2007; Burke, Elliman & Weare, 1994). According to Wang, Geng and Zhang (2019), GA is likely to gain high-quality approximation within a limited time compared to exact algorithm, especially on large-scale problems where the exact algorithm fails to take effect.

According to Ghazali and Ramli (2004), and Lambora, Gupta and Chopra (2019), the use of GA is already proven to be very successful by providing high efficient solution with less computation memory employment. Wang (2002) used the standard version of GA in solving lecturer assignment problem by aiming to maximize preferences of lecturers, to manage lecturer's load constraints and to overcome the variance in lecturers allocated hours. A penalty function must be given to the parameters that violate the constraints. From the study, it is found that GA could escape many inappropriate solutions and stimulate the search efficiency.

Another study done by Turki et.al (2014) used a special version of GA with the best detected parameter and evaluated operators by aiming to maximize lecturer's preferences, teaching quality, and the overall fairness in the assignment. The tournament selection method has been used. The proposed algorithm generates highly satisfied solutions in reasonable computing time.

Qi and Xu (2012) illustrated that GA able to optimize the effectiveness of lecturer assignment subject to specific department constraints. They also suggested that specific adjustments of algorithm and constraints are needed for different University. Besides, future work is required on detailed sensitivity studies of the developed algorithm and its performance on very large sets of lecturers and classes suitable for real applications.

While Gunawan, Ng and Ong (2008), and Aziz and Aizam (2018), solved the problem by formulating mathematical model to be applied in GA with two types of crossover. Comparison made in terms of computational results show that the proposed algorithm generates better solutions than manual assignment prepared by the department administrator. They proposed two phases of algorithm where the first phase concentrates on assigning the lecturers to the courses and set the number of courses to be assigned to each lecturer. Next, the second phase is to assign the lecturers to the course sections with the aims to balance their load.

Though, each of researchers used basic GA in their studies, none of the approaches are comparable in terms of the execution, for example parameters, processing techniques and problem encoding (Sultan, 2020).

2.7 Genetic Algorithm Generic Model

GA is an algorithm based on natural evolution process. The theory is inspired from Charles Darwin who proposed survival of the fittest. According to Lamini, Benhlima and Elbekri (2018), the basic GA comprises of genetic representation, fitness function and genetic operators. GA starts with a set of chromosomes, known as a population. A new solution formed from a set of population which is considered has a better fitness compared among other population. According to their environment, a parent (selected solution) then forms a new solution (offspring) that have higher chances to survive and reproduce. Offspring is produced through genetic operators which include cloning, crossover, and mutation. The process will keep repeated until it reach to a termination condition such as a number of iterations or a best solution obtained. According to Wang (2002), it is essential to transform problem into the parameter's optimization problem as to search a right combination of a series of parameters to obtain the most satisfaction either minimum or maximum, based on the requirement of the problem.

2.7.1 Genetic Operators

The genetic operators that produce new offspring are selection, crossover, and mutation:

a) Selection

Selection performs a significant role in leading towards optimal solution. The following are the two stages where the operator is used:

1) Selection of parents to create offspring before the crossover process. The fittest parents would produce fitter offspring.
2) Selection of fittest population for the next generation after manipulation of individual solutions.

This operator makes duplication of original chromosomes without altering their appearances. The probability that an element has been copied corresponds to its fitness. Among the popular selection types being used are tournament selection and roulette wheel selection. According to Goldberg and Deb (1991), the reasons why tournament selection becomes the most widely used selection method in GA are due to its efficiency and simple implementation. Chromosomes are selected randomly from a larger population and compete against each other. The chromosome with the highest fitness wins and then will be included in the next population. Roulette wheel selection is a genetic operator used in selecting potentially useful chromosomes for recombination. Chromosomes are selected with a probability that directly corresponds to their fitness values (Razali & Geraghty, 2011; Marzouk & Abdelakder, 2020). Other selection types available are rank selection, stochastic remainder and selection pressure.

b) Crossover

Crossover works by swapping the genetic material of parent chromosomes to generate offspring. For crossover, the parents are selected according to their fitness. Cheng and Gen (2019) highlighted that the crossover points could single point or two points. The offspring produced will become closer to the fittest solution in the population. The operator is shown in Figure 2.2.

Parents		Offsprings
P1 (01101**010**)	→	P1 (01101**000**)
P2 (11011**000**)		P2 (11011**010**)

Figure 2.2 Example of single point crossover operation

c) Mutation

Mutation helps to create variety in the search space. This operator helps in preventing the evolutionary process from getting trapped in local optima. The mutation operator is employed to chromosomes with its probability.

2.7.2 Parameters for Genetic Algorithms

Many researchers have performed experiments of how varying parameters affected the GA's search performance. Among the best parameter setting that has been widely used are as in Table 2.3.

Table 2.3 Parameter setting of GA described in the literature

No.	Researchers	Population size	Crossover rate	Mutation rate	Type of selection	Problem
1.	De Jong (1975)	50-100 individuals	~0.6 per pair of parents	0.001	Roulette wheel selection	General
2.	Grefenstette (1992)	20-30	0.75 – 0.95	0.005 – 0.01	Not stated	General

No.	Researchers	Population size	Crossover rate	Mutation rate	Type of selection	Problem
3.	Kazarlis, Petridis, and Fragkou (2005).	50	NA	0.001	Roulette Wheel Selection	UCTTP
4.	Nuntasen and Innet (2007)	42	0.93	0.07	Random	UCTTP
5.	Gunawan, Ng and Ong (2008)	100	0.75	0.05	Not stated	LAP
6.	David, Davies and Stanton (2013)	250	0.95	0.03	Roulette wheel selection	LAP
7.	Ahangaran, Pourbozorg and Talebi (2017)	512	0.5	0.1	Tournament	UCTTP
8.	Turki et.al (2014)	250	0.9	0.25	Tournament	LAP
9.	Gozali and Fujimura (2020)	30	0.8	0.1	Roulette wheel selection	UCTTP
10.	Puspitasari and Moengin (2020)	500	0.5	0.01	Stochastic Uniform	UCTTP

Notes:

1) UCTTP – University Course Timetabling Problem; 2) LAP – Lecturer assignment problem

Goldberg (1989) mentioned that a very small population size is better compared to large population. While studied done by Hassanat (2019) showed that larger population size is better. The suggested parameters were mutation rates (0.1), crossover rates (0.5) and range of population sizes (100, 300, 400 and 600). They found that the accuracy of the GA increased, more chances for the initial population comprise of individual presenting the optimal solution and also the increased of number of generations' convergence.

According to experiments done by Berisha, Bytyciand Tershnjaku (2017), the tournament selection method allows for very quick solutions in light-constrained problems, using a fine-grained parallelism method. While roulette selection can be used in a coarse-grained parallel algorithm with slightly slower but better results in even harder problems. Therefore, the advantage of the model is in cases of bigger population.

2.8 Summary

University timetabling problem is categorized as NP-hard problem which dealing with assigning courses to time slot and classroom by trying to fulfil the preferences of lecturers and satisfy all the constraints. From past studies, approaches to solve this problem can be categorized into three i.e. exact, metaheuristic and intelligent novels. GA as being categorized as metaheuristic is chosen for this study because of its vast successful applications to combinatorial optimization problems and NP-hard problems. Also, it has the potential of finding an optimal solution. The identified constraints based on the previous studies such as considering day and lecturers' time preferences will be adapted for this study. The methodology involved in GA will further discuss in Chapter 3.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter explain the methodology that is used to achieve the objective of this study. Section 3.2 begins with the illustration of the flow of overall research design as to achieve each of the stated objectives. Section 3.3 discusses the problem definition of the study; follow by discussion of the data related to course timetabling and classroom assignment problem in Section 3.4. Later Section 3.5 discusses the constraints involved and the formulation of mathematical model. The following Section 3.6 and 3.7 mainly explain the steps involved in the proposed heuristics using GA and how the model will be validated and verified. This chapter ends with expected finding in Section 3.8 and summary of the proposed methodology in solving a combination of course timetabling and classroom assignment ptroblem.

3.2 Research Design

This research aims to develop a model for a combination of course timetabling and classroom assignment problem. To achieve the three listed objectives, this research is conducted through five phases of research activities: problem formulation, data collection, mathematical formulation, model development (the proposed heuristics), and model validation and verification.

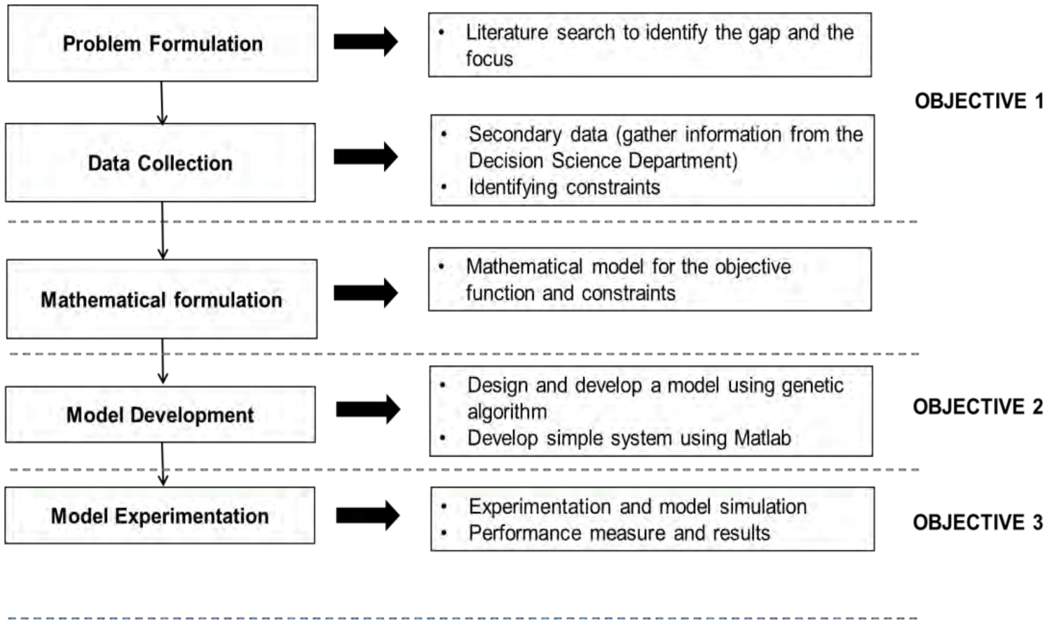


Figure 3.1 Flow chart of research phases

3.3 Problem Formulation

The main concern of this study was to assign the lecturers, courses and course sections to time slots and classroom. Subject taught by a lecturer in a week is referred as a course. Each course has few sections and need to be assigned to day, time slots and classroom. The problem was formulated based on literature review and real problem of lecturer assignment in the Department of Decision Science, SQS, UUM.

3.4 Data Collection

Two types of data, primary data and secondary data were collected for this study. The primary data was gathered from the interview sessions, while the secondary data consist of different types of dataset including lecturers' teaching preference for course day and time slot, and list of available classroom. All these data became the constraints and preferences to the proposed algorithm.

3.4.1 Primary data

Primary data was collected through short interview with undergraduate and postgraduate programme coordinator from Decision Science Department, School of Quantitative Sciences (SQS). The data gave an overview and information on how the current practice or system of course timetabling and classroom assignment being implemented.

At the time this study being conducted, the task of allocating courses to the lecturers was being done manually by the undergraduate and postgraduate programme's coordinator as the number of staff can be considered small. However, the task became more complicated as the number of lecturers increases. Each lecturer was asked to fill in a course selection form.

3.4.2 Secondary data

The data set that was collected from a total of 27 available lecturers under Decision Science Department for Semester A172. The method requires the inputs in three excel sheet table. Table 3.1 should include the course code, course hour, lecturer name and number of students registered. In addition to that fields, input for courses preferred day, classroom, starting hour and effect of selection need to be filled.

27 lecturers were allocated to teach 17 courses being offered for Semester A172. The numbers of class sections for each course are as depicted in Table 3.1.

Table 3.1 List of courses and number of class section offered for Semester A172

	COURSE NAME	NUMBER OF CLASS SECTIONS
1.	**SQIT1013** *Pengaturcaraan Dalam Aplikasi Perniagaan*	**4**
2.	**SQIT3014** *Pembuatan Keputusan Berbantu Komputer*	**4**
3.	**SQIT3033** *Perolehan Pengetahuan Dalam Pembuatan Keputusan*	**3**
4.	**SQIT3043** *Permodelan Data Dan Pangkalan Data*	**1**
5.	**SQIT5013** *Business Programming Using Visual Tools*	**1**
6.	**SQQP1014** *Teknik Sains Kuantitatif I*	**1**
7.	**SQQP2014** *Teknik Sains Kuantitatif II*	**5**
8.	**SQQP3023** *Pemodelan Pemutusan*	**2**
9.	**SQQP3033** *Pemodelan Berkomputer Dalam Perniagaan*	**2**
10.	**SQQP3043** *Teknik-Teknik Heuristik*	**2**
11.	**SQQP3063** *Pemodelan Sistem Dinamik*	**2**
12.	**SQQP3073** *Penyelidikan Dalam Sains Kuantitatif I*	**9**
13.	**SQQP4073** *Penyelidikan Dalam Sains Kuantitatif II*	**1**
14.	**SQQP5043** Simulation for Decision Making	**1**
15.	**SQQP6023** Heuristic Technique for Combinatorial Optimization Problems	**1**
16.	**SQQS3183** *Topik Khas Dalam Statistik*	**1**
17.	**SQQP6014** *Operation Research*	**1**
	Grand Total	**41**

Table 3.2 is the first sample of input in Excel (Sheet 1). According to the sample, the first-row course and third-row course had some selection. First-row

course's preferred classroom id is 4, the preferred day is on Monday (2), while the preferred hour is zero means there is no selected time slot. If the algorithm assigns that course to the preferred classroom and day, therefore it would gain 0.5 points per student who registered that course. For second row, the values for all selection were zeros which means there is no selection for preferred classroom, day and hour.

Table 3.2 Sample of input data for each course code

Course Code	Course Name	Course Section	Hour	Credit	Lecturer ID	Number of Students	Preferred classroom [1...13]	Preferred day [1...5]	Preferred time slot [1…14]	Effect of Selection [0...1]
SQIT1013	*Pengaturcaraan Dalam Aplikasi Perniagaan*	A	**4**	**4**	1234	20	4	2	0	0.5
SQIT1013	*Pengaturcaraan Dalam Aplikasi Perniagaan*	C	**3**	**3**	1235	20	0	0	0	0
SQIT3033	*Perolehan Pengetahuan Dalam Pembuatan Keputusan*	D	**2**	**2**	1236	20	3	3	1	0.9

Table 3.3 is the division of time slots for five days and their penalty values per student if we assign any course at the certain time slot. This input was available in Sheet 2 of Excel. The first time slot penalty is 0.5 and the last time slot penalty is 1 per student. The penalty given as we do not prefer to assign classes early in the morning and late evening.

Lectures were held from Sunday to Thursday with 14 time slots with 45 minutes each. Lectures start at 08:00-08.45 to 17:45-18:30.

Table 3.3 Division of time slots

Time Slots		Day 1	Day 2	Day 3	Day 4	Day 5
Slot 1	08:00-08:45	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**
Slot 2	08:45-09:30	0	0	0	0	0
Slot 3	09:30-10:15	0	0	0	0	0
Slot 4	10:15-11:00	0	0	0	0	0
Slot 5	11:00-11:45	0	0	0	0	0
Slot 6	11:45-12:30	0	0	0	0	0
Slot 7	12:30-13:15	0	0	0	0	0
Slot 8	13:15-14:00	0	0	0	0	0
Slot 9	14:00-14:45	0	0	0	0	0
Slot 10	14:45-15:30	0	0	0	0	0
Slot 11	15:30-16:15	0	0	0	0	0
Slot 12	16:15-17:00	0	0	0	0	0
Slot 13	17:00-17:45	0	0	0	0	0
Slot 14	17:45-18:30	**1**	**1**	**1**	**1**	**1**

While Table 3.4 is the list of classrooms and capacities for each. This input data is the available in Sheet 3 of Excel.

Table 3.4 List of available classrooms

No.	Classroom	Capacities
1	DKG 6/5	120
2	SQS BT1	30
3	SQS BT11	30
4	SQS BT3	30
5	SQS BT5	30
6	SQS BT7	30
7	SQS BT9	30
8	SQS DP1	100
9	SQS DP3	100
10	SQS Lab 1	40
11	SQS Lab 2	40
12	SQS Lab 3	40
13	SQS Lab 4	40

3.5 Mathematical Formulation

Related issues are to be addressed as high priority issues, which reflects the urgency to be fulfilled and the low priority issues are the side-line matters to contend with. The constraints involved are the hard constraints and soft constraints. The hard constraints will focus on the mandatory specifications which must be fulfilled and cannot be violated, while the soft constraints would be integrated in the objective function that require to be maximized. The soft constraints are normally referred as preferences to lecturer's demand or conditions required for the timetable as preferences. It associates with a weight or preference value, and level of preferences and importance. Mathematical formulation was formulated based on identified constraints in this study.

The common requirements are as follows:

1. Lecturers cannot teach two courses at the same time (Hard constraint).
2. One room for one course at one time (Hard constraint).
3. All course section should be assigned to classroom that could be sufficiently accommodate the student number (Hard constraint).
4. Preferences in terms of preferred day, time slot and classroom are maximized (Soft constraint).
5. Each course must run continuously without any breaks. This means that courses with a weight of 2 credits must fill 2 time slots continuously, while for courses weighing 3 credits, they must fill 3 time slots continuously (Soft constraint).

Assumptions:

The lecturers from Decision Science (DS) Department were assumed only to teach the courses offered from DS programmes. The assignment problem considered in this study is carried out under the condition that the lecturer, courses and section have been arranged previously.

In this study, the mathematical model used for this problem was adopted from Puspitasari and Moengin (2020). The fitness function of the scheduling problem used in this study is maximizing the preference of the teaching lecturer at the preferred hour, day and room, maximizing the room capacity for each course and minimizing the courses assigned at an unwanted time. Next, the fitness function and absolute constraint are modelled into a mathematical equation. The following nomenclatures will be used throughout the report:

n : Number of courses (including parallelized ones)

R : Set of real numbers

L : Set of lecturers

D : Set of days (represented in $\{1,2,3,4,5\}$)

T : Set of time slots per day (represented in $\{1,2,..,14\}$)

C : Set of classroom (represented in $\{1,2,..,13\}$)

M : Set of course codes

S : Set of course IDs (represented in $\{1,\ldots,n\}$)

$m_1, m_2, \ldots m_{|s|} \in M$: Course code

$l_1, l_2, \ldots l_{|s|} \in L$: Lecturer for the sth course id, $s \in S$

$a_1, a_2, \ldots a_{|s|} \in R$	:	Number of classroom capacities
$b_1, b_2, \ldots b_{|s|} \in R$	:	Total credit hours
$cp_1, cp_2, \ldots cp_{|s|} \in C$	:	Lecturer's preference on classroom for the s^{th} course id
$dp_1, dp_2, \ldots dp_{|s|} \in D$	:	Lecturer's preference on day for the s^{th} course id
$tp_1, tp_2, \ldots tp_{|s|} \in T$	:	Lecturer's preference on time slot for the s^{th} course id
$ep_1, ep_2, \ldots ep_{|s|} \in R$	:	The gain obtained by the course when assigned to the room, day and time slot desired by the lecturer
$P_{d,t} \in R$	:	Penalty earned by the course when assigned on the d, day and t, time slot
$ct_1, ct_2, \ldots ct_{|c|} \in C$	:	Penalty earned by the course when assigned on the d, day and t, time slot

Define $X_{d,t,c,s} \in \{0,1\}$, where $X_{d,t,c,s} = 1$ if course ids was assigned on day d, time slot t, and room c, and $X_{d,t,c,s} = 0$ otherwise. So that the course timetabling problem could be modelled into the following mathematical equation:

$$\text{Min } Z = -\sum_{i=1}^{|s|} A_i . Ep_i . X_{Dpi.Tpi.Cpi} - \sum_{d=1}^{|D|} . \sum_{t=1}^{|T|} . \sum_{c=1}^{|C|} . \sum_{s=1}^{|S|} (Ct_{c-} A_s) . X_{dtcs} + \sum_{d=1}^{|D|} . \sum_{t=1}^{|T|} . \sum_{c=1}^{|C|} . \sum_{s=1}^{|S|} P . A_s . X_{dtcs}$$

(Equation 1)

Constraint 1: A lecturer cannot be assigned more than one course at the same time

$$\sum_{c=1}^{|C|} . \sum_{s=1}^{|S|} X_{dtcs} = 1 \text{ if } L_s = L_l, \{\forall s \in S, \forall l \in L, \forall d \in D, \forall t \in T\}$$

(Equation 2)

Constraint 2: A classroom must be assigned only one course at the same time

$\sum_{s=1}^{|S|} X_{dtcs} = 1$ $\{\forall d \in D, \forall t \in T, \forall c \in C\}$

(Equation 3)

Constraint 3: Number of students enrol for each course must be less or equal to the capacity of the classroom

$X_{dtcs} . A_s \leq Ct_C$, $\{\forall d \in D, \forall t \in T, \forall c \in C, \forall s \in S\}$ (Equation 4)

Constraint 4: All time slots must be assigned to courses continuously without any break

$$\sum_{o=j+1}^{j+Bs-1} X_{dtcs} = B_i - 1 , \quad if \begin{pmatrix} j = starting\ hour\ of\ course \\ d = course\ day \\ c = course\ classroom \end{pmatrix} \{\forall s \in S\}$$

(Equation 5)

3.6 Model Development for Assigning Lecturer to Course Timetabling and Classroom

The problem was solved using GA as discussed previously. Following is the pseudocode of the GA method:

1) Define fitness function
2) N=stopping criteria
3) Chromosome representation
4) Initialization of chromosomes in the population randomly
5) Evaluation of chromosomes in the population
6) t=0

```
while t
{
        do
    a.  Chromosomal selection
    b.  Chromosomal crossover
    c.  Chromosomal mutation
    d.  General Replacement
t=t+1
}
```

The steps of these method is shown in Figure 3.2.

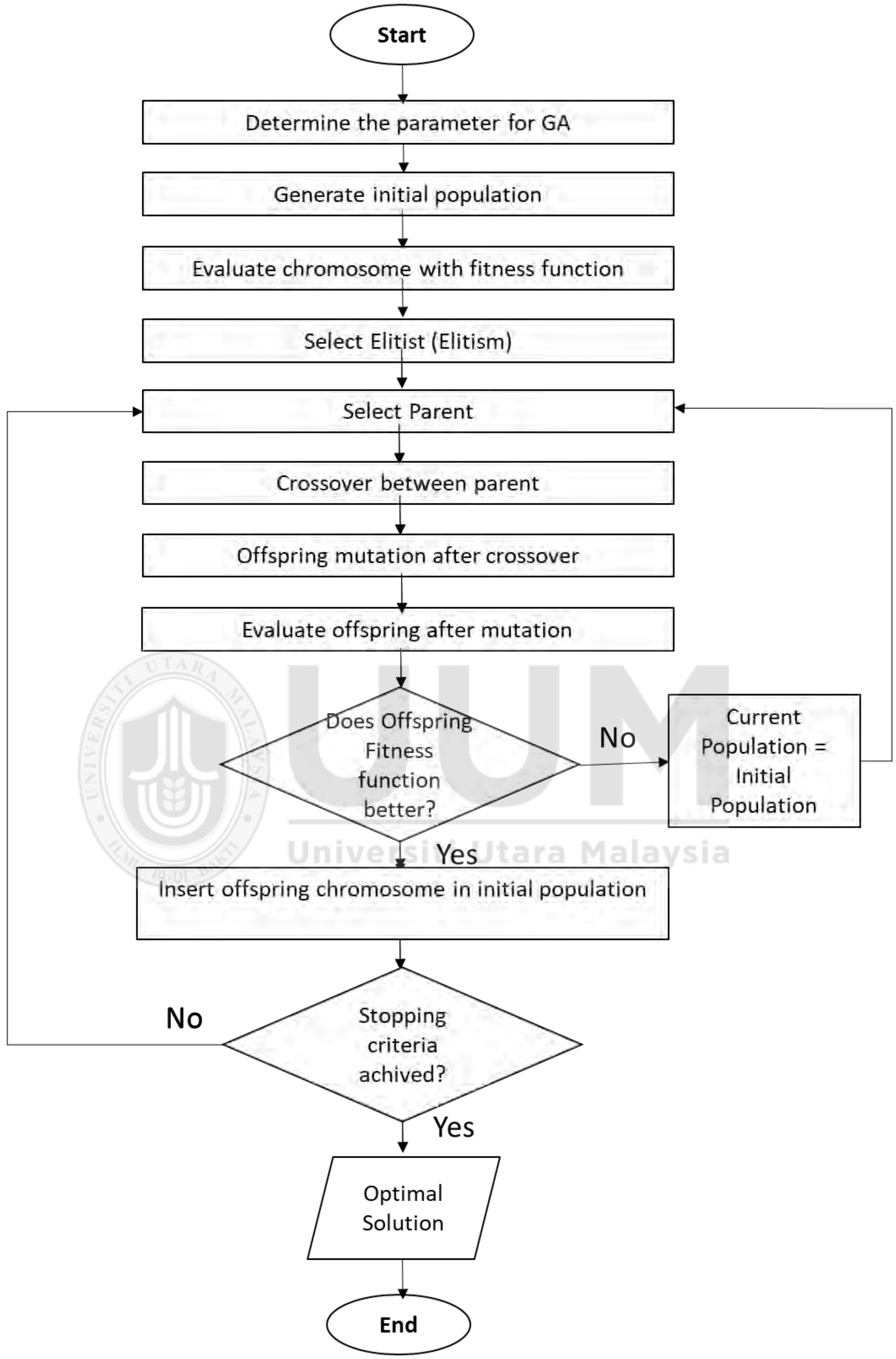


Figure 3.2 The proposed flowchart in assigning courses

a) Parameter

Table 3.5 is the list of variables, while Table 3.6 is the parameter used for running the test simulation. In this study, basic parameter selection is due to the simulation of several combinations of parameter values. The parameter for mutation rate, population, iteration and crossover was adopted from Puspitasari and Moengin (2020).

Table 3.5 The Variable Values

Variable	Total
Number of time slots	14
Number of rooms	13
Number of courses	17
Number of course section	41
Number of lecturers	27

The experimentation to find the best parameter setting was determined based on combination of different mutation rate, population and iteration as shown in Table 3.6.

Table 3.6 The Parameter Values

Parameter	Value
Mutation rate	1%- 5%
Population number	100 – 500
Iteration number	100 – 200
Crossover type	Scattered

b) Population initialization

Course matrices, clashing matrices, matrices lecturer's preferences, time slot matrix, matrix room, and initial assignment matrix used as input to calculate the number of possibilities of all courses to be assigned on the day, time slot and the room that fills the entire hard constraint.

The clashing matrix is $M_{n \times n}$ where n is the number of courses to be scheduled. The value clashing matrix element binary, has a value of 1 if 'clashed' and a value of 0 if 'no clash'.

c) Initial Population Evaluation

After all courses have calculated the possible assigns based on the chromosome value of the initial population, the next step is to calculate the fitness value of each of the selected assigns by making the corresponding chromosome value 1 if it matches the preference, while the one that does not match the preference is 0. The population evaluation is calculated by fitness function based on equation 1.

d) Elitism Population

The elitism process is carried out by copying the best chromosomes so that they are not lost during the crossover and mutation process. The elitism rate used is 5%, meaning that the five best chromosomes are selected from the total N number of chromosomes, and copied into the population.

e) Selection

Then, the parent selection was carried out by selecting two chromosomes using the stochastic uniform method. These two chromosomes will be the parent to carry out the crossover process.

f) Crossover

Crossover in this study is done by scattered crossover. The results of the crossover of genes from the parent will be passed on to the offspring.

g) Mutation

The next stage is to carry out the mutation process by using random mutation for the offspring obtained from the crossover process. The mutation process is by replacing the gene value of a chromosome with a value obtained at random.

h) Offspring Chromosomes Evaluation

After the mutation process is carried out, if the fitness value of the mutated offspring's chromosome is better, it will enter the initial population to replace the worst chromosome in the population, whereas if the mutated chromosome is bad, it will be discarded and the current population remains the same as the initial population.

i) Stopping Criteria

The process from parent selection to mutation is referred to as one generation, then the population will continue to generate until one of the stopping criteria is reached, i.e. when the number of iterations is maximum or when the best solution is found.

j) Best Solution

The GA method can quickly reach local minimum values, but requires a lot of function evaluation to achieve convergence. After the stopping criteria are met, an output in the form of an optimal chromosome will be produced which will produce the best fitness value when multiplied by the possibility of assigning all courses. The GA system was implemented using MATLAB (R2018a) and simulations being implemented using Intel Core i5 processor with 4GB of memory and Window 10. The parameters of GA's performance were based from the literature review such as number of assignment generated, initial population size, number of generation for GA and mutation rate.

In brief, the proposed MATLAB consists of three modules which are:

i. Input lecturer and course details

   User is able to input the lecturer's preferences into Excel file (Sheet 1) that was imported to MATLAB file so that the system could begin allocating lecturers based on Sheet 2 (division of time) and Sheet 3 (classroom).

ii. Allocate lecturers and courses to days, time slots and classrooms

   A search algorithm using GA was created to assign lecturers, courses and course sections to days, time slots and classroom

iii. Display result

   The course timetabling was displayed in Excel output obtained from MATLAB.

3.7 Model Experimentation

Model experimentation was the last phase in research methodology. This part involved testing and verification activities to measure the performance of a proposed approach and to find the optimal objective function for the lecturer assignment problem by varying the parameter values. Comparison of the results is to examine the effects of the best solution. As the problem was minimization problem, solution comparisons were made by taking into account the value of the more negative fitness function.

3.8 Summary

This chapter discusses on the methodology of the GA in solving university timetabling problem. Three objectives has been stated together with the steps involves in achieving the objectives. The first objective which is to identify the requirements of university timetabling problem that needs to be satisfied in generating feasible timetable has been achieved at Section 3.4 where the required table has been displayed and discussed. While the second objective which is to develop a GA model for solving university timetabling problem that satisfying constraints has been achieved at Section 3.6. Finally, third objective which is the result of the best parameter of GA in solving the University timetabling problem will be discussed at Chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter presents the findings and the discussion of this study. Model experimentations of the proposed algorithm are discussed in this chapter. These activities were carried out to evaluate the performance of the algorithms. All experimentations were carried out using real data provided by undergraduate and postgraduate course coordinator.

4.2 Computational Experiment

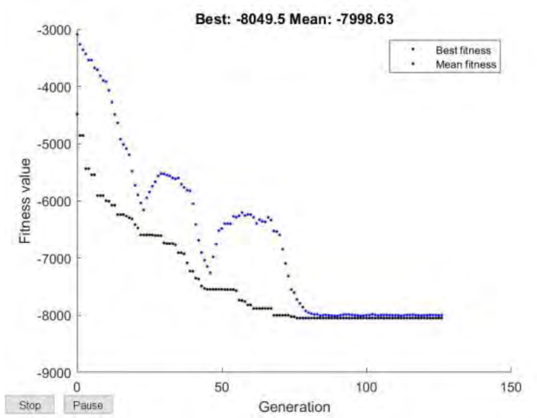
Table 4.1 shows the comparison of result for 27 simulations with varied parameter values of population and iteration as being mentioned in Table 3.7 (parameter list). The results consist of fitness value and graph of best fitness and mean fitness versus generation.

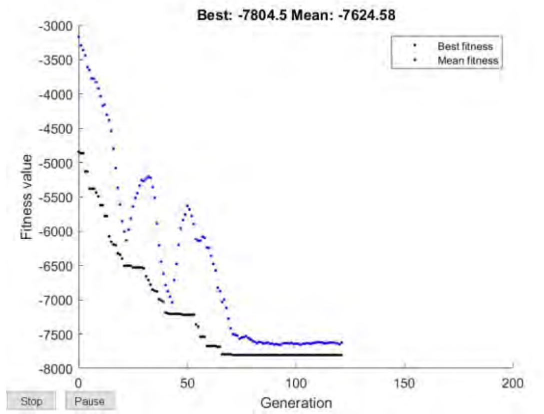
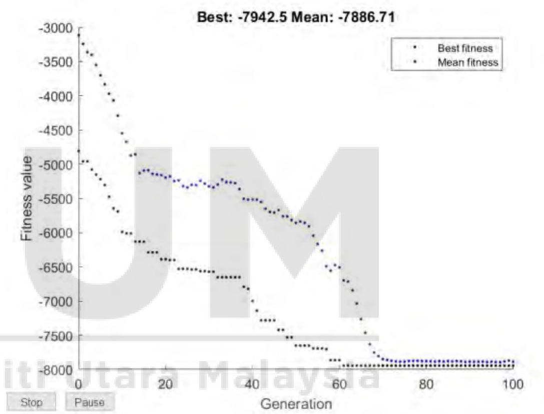
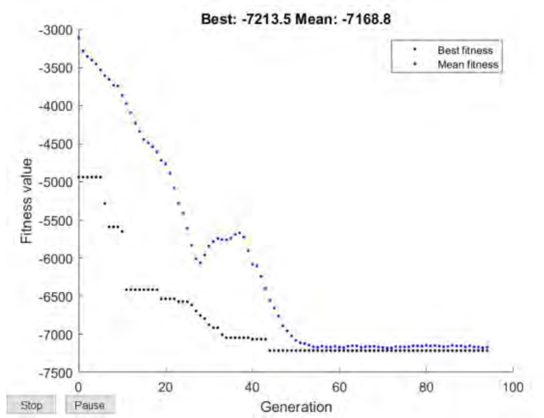
Best fitness is referring to either the fitness of the best individual during the test run or the fitness of the best individual in the current population. While the mean fitness is simply the average of the fitness values across the entire population. Each generation, the population changes and you get a new average population fitness. Generally, the best fitness is the best as we only need one best solution. Mean fitness is beneficial in displaying the algorithm activities. If the average fitness reaches the best fitness, it likely means that the population contains N copies of the best individual and further search is likely to be wasted effort without a restart or large-scale mutation operation.

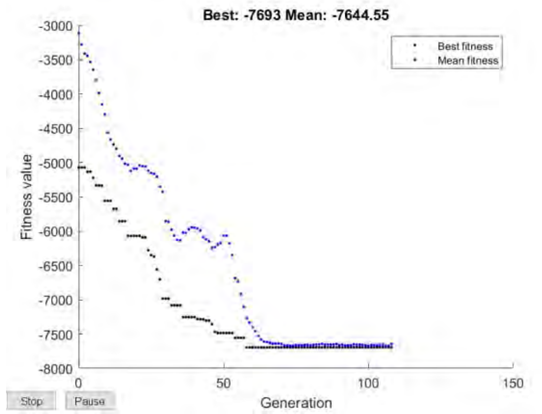
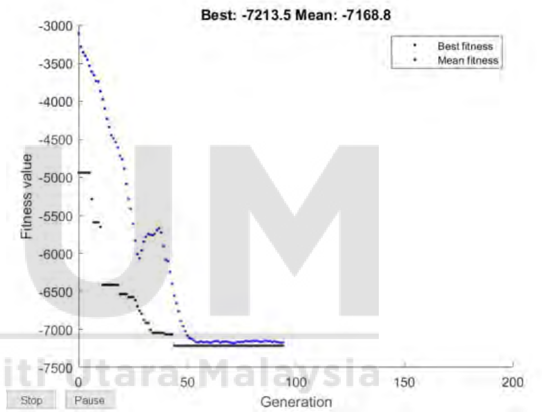
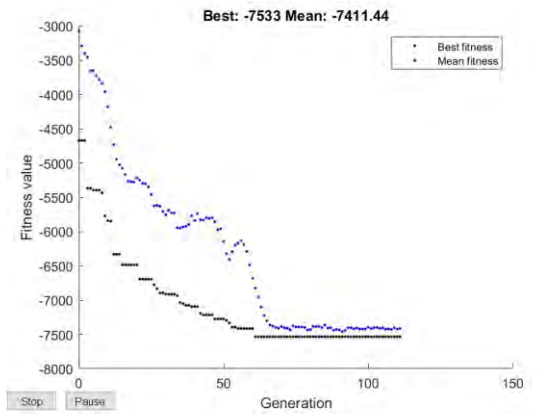
Best fitness tends to get improve over time, rapidly at first, and then decelerating as the algorithm finds better and better solutions that are harder to improve upon. While the mean fitness is usually somewhat not good as best fitness, with the difference between the two typically decreasing over time till the algorithm converges totally (the population is uniform, containing N copies of the same "best" individual).

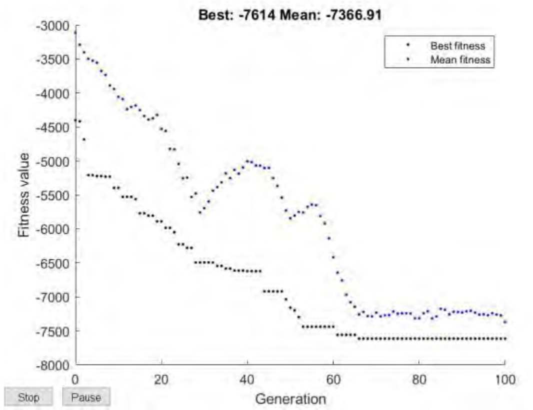
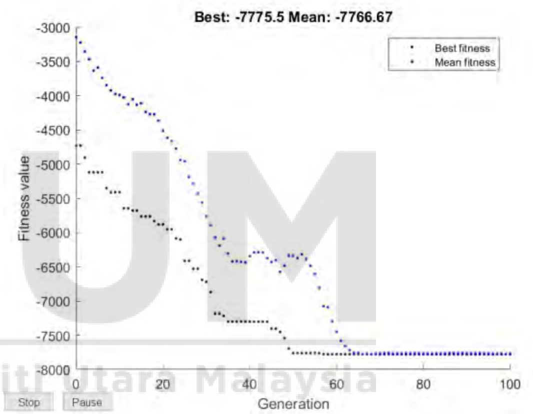
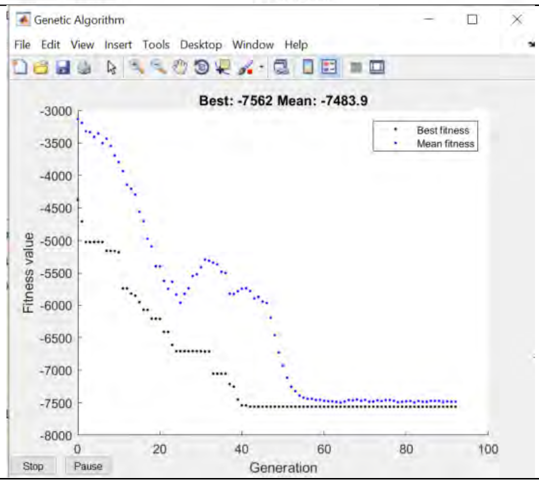
The simulation results in Table 4.1 involved 27 different combinations of parameter values, such as changing the value of population (500, 400, 300 and 200), iteration (100, 150 and 200) and mutation rate (0.1, 0.3 and 0.5). While the elitism rate was set 5% based on suggested in literature review.

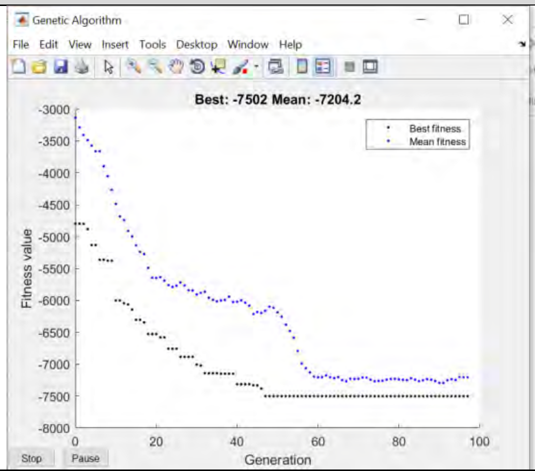
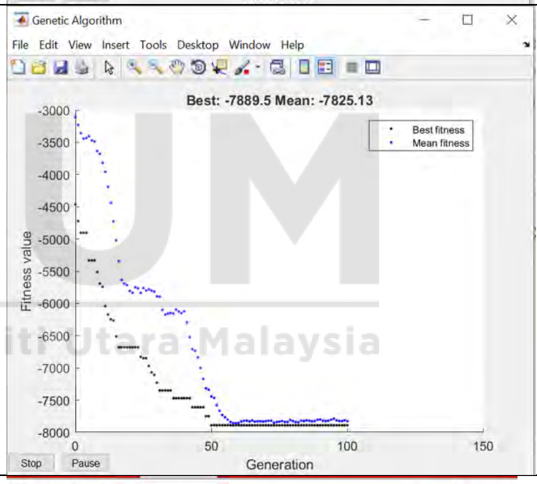
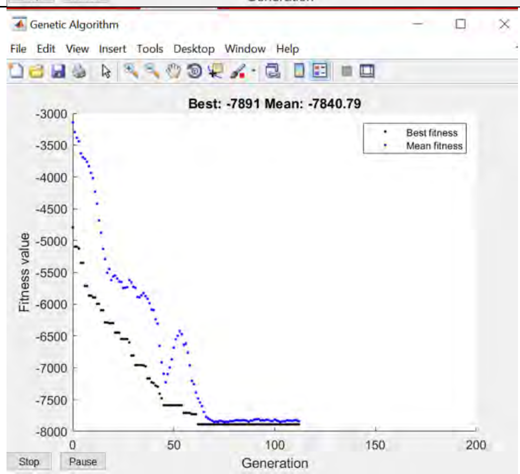
Table 4.1 Comparison of result for 27 simulations

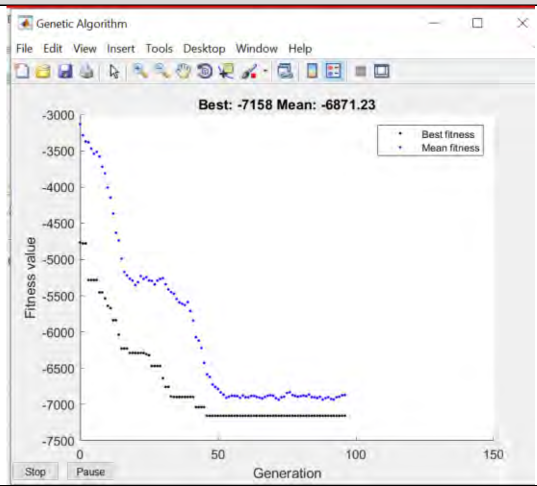
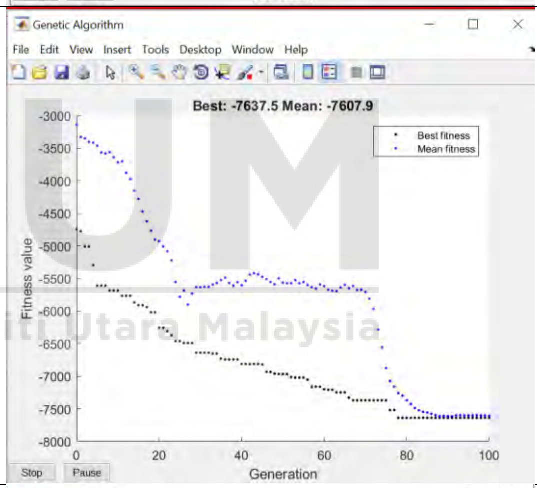
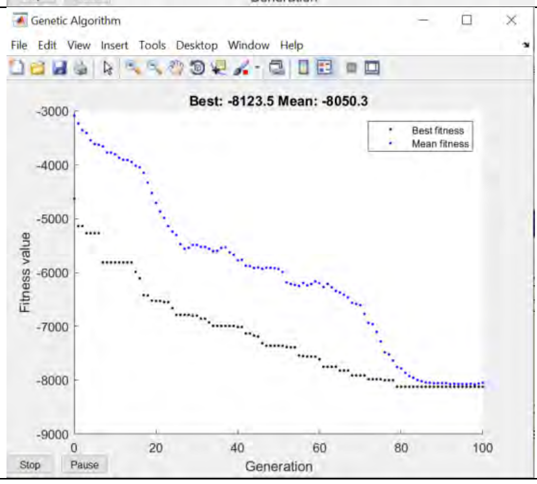
Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
1	500	150	0.1	1.005888		-8489.5 (Generation : 126)

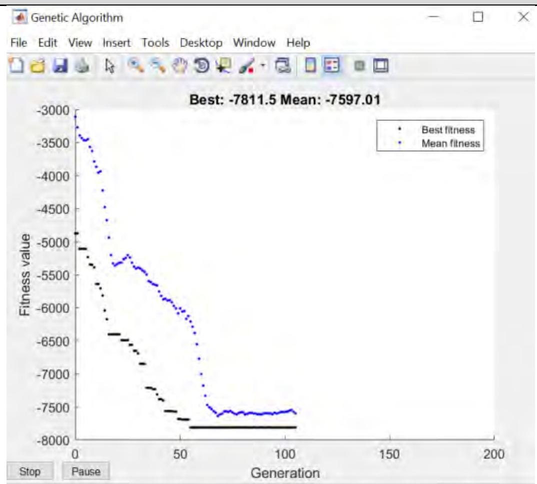
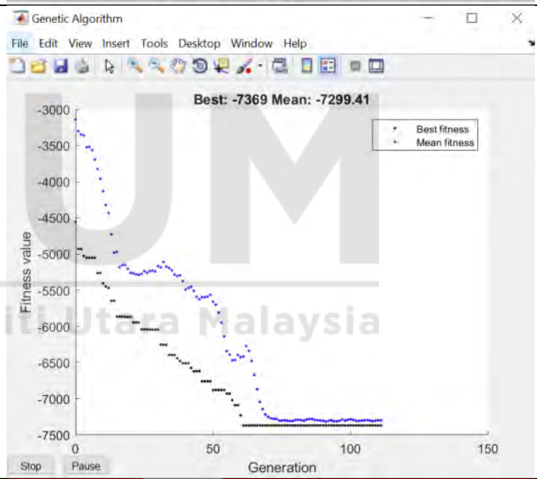
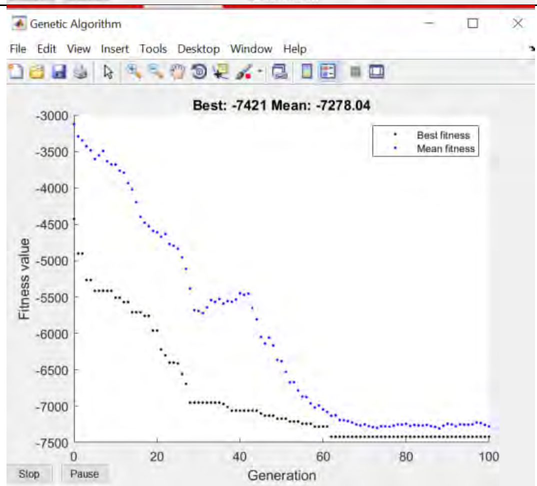
Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
2	500	200	0.1	1.021998		-7964.5 (Generation : 121)
3	500	100	0.1	1.062250		-8342.5 (Generation : 100)
4	400	100	0.1	1.014508		-7658.5 (Generation : 94)

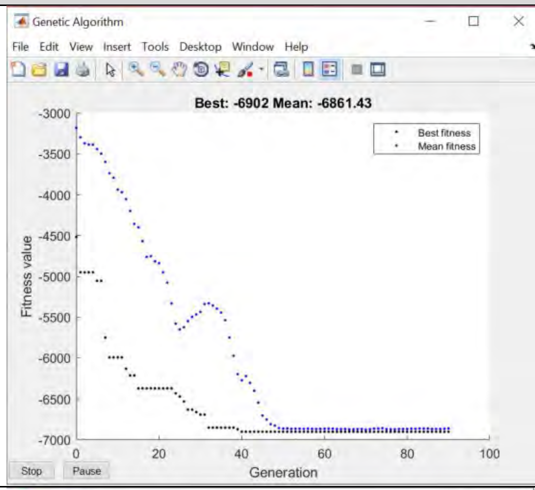
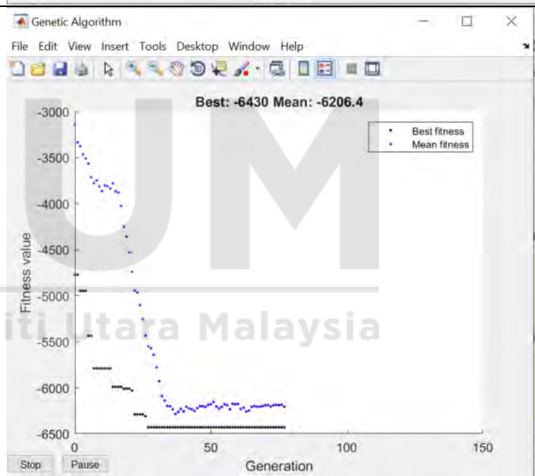
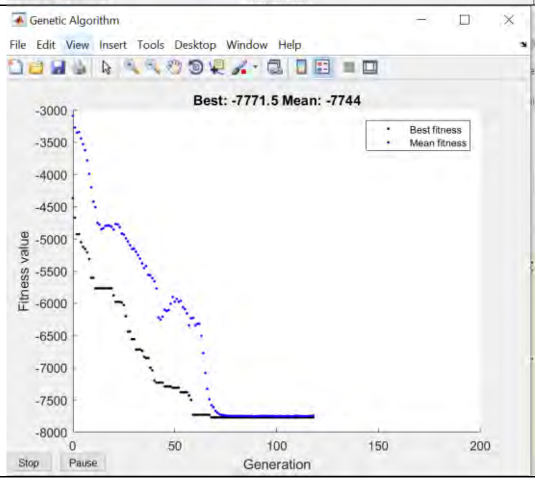
Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
5	400	150	0.1	1.014006		-8113 (Generation : 108)
6	400	200	0.1	1.003437		-7658.5 (Generation : 94)
7	300	150	0.1	1.004681		-8806 (Generation : 111)

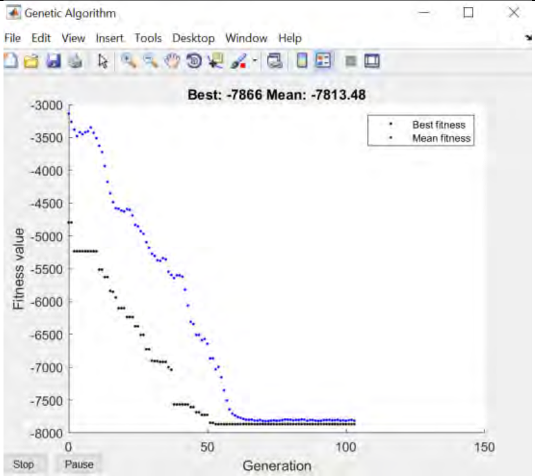
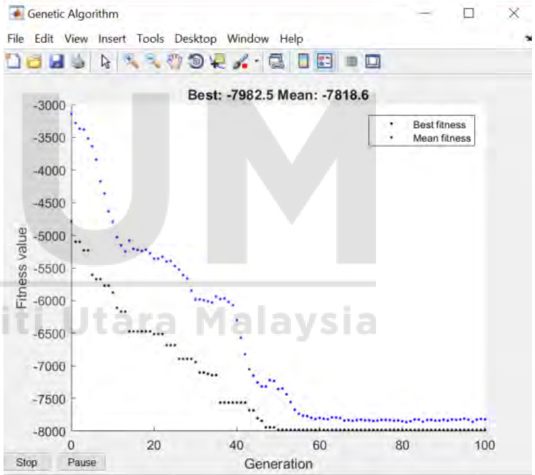
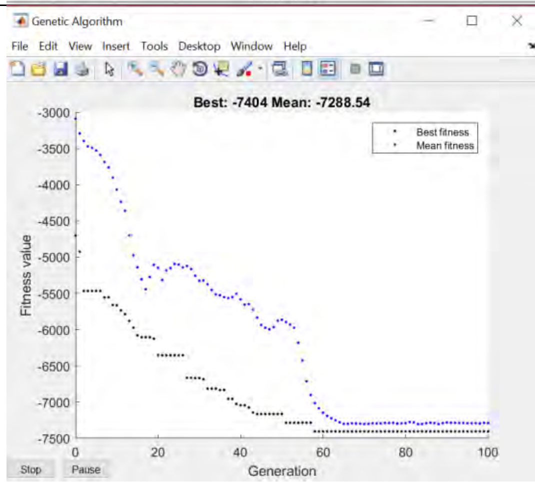
Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
8	300	100	0.1	1.000903		-8014 (Generation : 100)
9	200	100	0.1	1.031862		-8275.5 (Generation : 100)
10	200	100	0.5	1.118910		-8086 (Generation : 92)

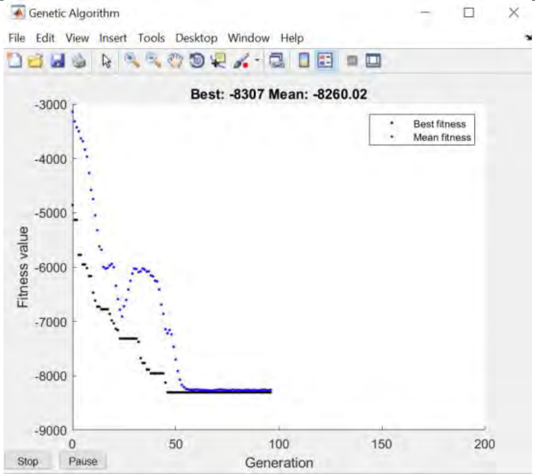
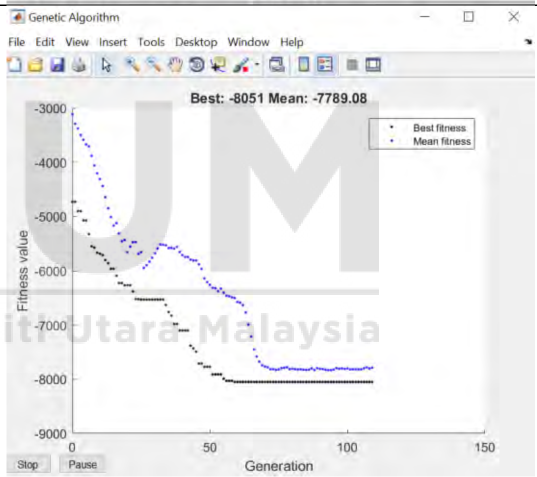
Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
11	300	100	0.5	1.004232		-8032 (Generation : 97)
12	300	150	0.5	1.005576		-8357 (Generation : 100)
13	400	200	0.5	1.00693		-8389 (Generation : 112)

Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
14	400	150	0.5	1.003389		-7938 (Generation : 96)
15	400	100	0.5	1.002557		-7967 (Generation : 100)
16	500	100	0.5	1.003543		-8655 (Generation : 100)

Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
17	500	200	0.5	999966666		-8871.5 (Generation : 105)
18	500	150	0.5	1.003116		-7719 (Generation : 111)
19	200	100	0.3	1.007824		-7671 (Generation : 1090)

Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
20	300	100	0.3	1.009320		-8174 (Generation : 90)
21	300	150	0.3	1.002908		-7520 (Generation : 77)9
22	400	200	0.3	1.003236		-8369.5 (Generation : 118)

Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
23	400	150	0.3	1.003091		-7931 (Generation : 105)
24	400	100	0.3	1.005009		-8254.5 (Generation : 100)
25	500	100	0.3	1.008844		-8004 (Generation : 100)

Test Run No.	Parameter			Time (Sec)	Graph of Best Fitness and Mean Fitness vs Generation	Best Fitness Value (Generation)
	Population	Iteration	Mutation rate			
26	500	200	0.3	1.001067		-8477 (Generation : 96)
27	500	150	0.3	1.013586		-8051 (Generation : 109)

After 27 simulations have been performed by varying the parameter values, the results of the running program were shown in Table 4.1 with the most negative number was found at test run-17. It can be seen combination of parameters (population (500); iteration (200) and mutation rate (0.5)) gave the best performance by giving the average best fitness value, -8871.5 and successfully fulfilled all the soft and hard constraints. For test run-17, the population continued to generate until one of the stopping criteria was reached at 105, which happened when the number of iterations was maximum or when the best solution was found. It took 1.006958 seconds to generate the timetable with 100% of the 41 courses were successfully assigned. The fitness function of the solution is negative because the fitness function is in the form of minimization, meaning that the more negative the value of the fitness function, the better it represents a solution that satisfies every constraint. Meanwhile, test run-7 and test run-16 were placed at the 2nd and 3rd ranked for the most negative best fitness that contribute to 95% and 85% of feasible courses assigned respectively.

The population size was the control parameter that had the most significant effect among the other control parameters. The large population sizes show better results than the smaller population size where population valued 500 led to the most negative fitness value. While, other parameters did not gives much effect. It can be seen in Figure 4.1, which shows the population versus best fitness. While, Figure 4.2 is the comparison of test run versus time in seconds. The fastest computational time to generate result is 1.000903 seconds from test run-8 and the longest computational time taken by test run-10 (1.1891 seconds). The computation time did not give any difference with the fitness value obtained.

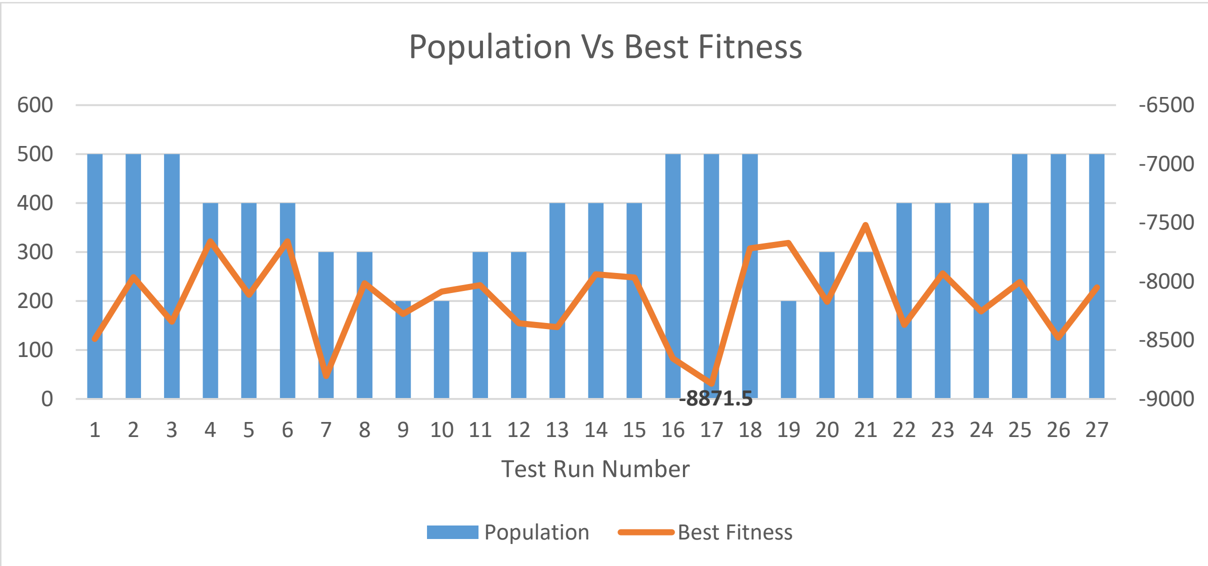


Figure 4.1 Population Vs Best Fitness

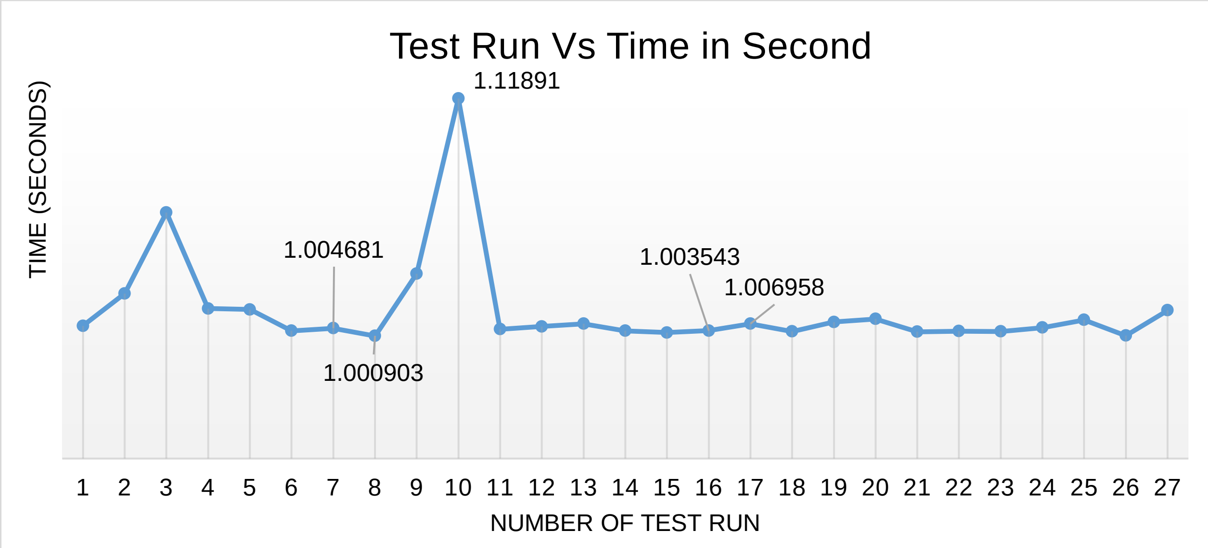


Figure 4.2 Comparison of computational time

According to the best fitness value achieved at test run-17, the final timetable generated in Matlab excel output show on the following Table 4.2. In this table, courses with a weight of two (2) credits fill two (2) time slots (95 minutes) continuously, while courses with a weight of three (3) credits fill three (3) time slots (145 minutes) and four (4) credits fill four (4) time slots (190 minutes) continuously.

Table 4.2 A sample solution of class schedule generated from the best solution (test run-17)

Slot number	Day 1 - Sunday	Day 2 - Monday	Day 3 - Tuesday	Day 4 – Wednesday	Day 5 – Thursday
Slot 1			SQQP3073-E-(2190)-SQS DP3		SQQP2014-C- (2095)-DKG 6/5 SQQP2014-B- (1607)-SQS DP1
Slot 2		SQIT1013-D- (2313)-SQS DP1	SQQP3043-B - (2186)-DKG	SQQP1014-A- (2193)-	SQQP2014-C- (2095)-

Slot number	Day 1 - Sunday	Day 2 - Monday	Day 3 - Tuesday	Day 4 – Wednesday	Day 5 – Thursday
			6/5 SQQP3073-E-(2190)-SQS DP3	SQS DP1 SQIT3014-C- (2311)-SQS DP3	DKG 6/5 SQQP2014-B- (1607)-SQS DP1
Slot 3	SQQP2014-E- (2095)-DKG 6/5 SQIT5013 -A- (2145)-SQS DP1	SQIT1013-D- (2313)-SQS DP1 SQQP3023-A- (2193)-SQS DP3	SQQP3043-B - (2186)-DKG 6/5	SQQP1014-A- (2193)-SQS DP1 SQIT3014-C- (2311)-SQS DP3	SQQP2014-C- (2095)-DKG 6/5 SQQP2014-B- (1607)-SQS DP1 SQIT1013-B- (2145)-SQS DP3
Slot 4	SQQP2014-E- (2095)-DKG 6/5 SQIT5013 -A- (2145)-SQS DP1	SQIT1013-D- (2313)-SQS DP1 SQQP3023-A- (2193)-SQS DP3	SQQP5043 -A- (2152) -DKG 6/5 SQQP3043-A-(2149)-SQS DP3	SQQP1014-A- (2193)-SQS DP1 SQIT3014-C- (2311)-SQS DP3	ENM400-H-(1429)-DKG 6/5 SQIT1013-B- (2145)-SQS DP3
Slot 5		SQQP3023-B- (2064)-DKG 6/5 SQIT1013-D- (2313)-SQS DP1	SQQP5043 -A- (2152) -DKG 6/5 SQQP3043-A-(2149)-SQS DP3	SQQP1014-A- (2193)-SQS DP1 SQIT3014-C- (2311)-SQS DP3	ENM400-H-(1429)-DKG 6/5 SQQP3033-A- (2158)-SQS DP1 SQIT1013-B- (2145)-SQS DP3
Slot 6	SQIT3014-B- (2320)-DKG 6/5 SQQP4073-A- (2157)-SQS DP3	SQQP3023-B- (2064)-DKG 6/5 SQQP3073-H- (2152)-SQS DP1 SQQP3073-C- (2190)-SQS DP3 SQQP3073-B- (2243)-SQS L4	SQQP3063-B-(2966)/(2304)-DKG 6/5 SQQP3063-A-(2157)-SQS DP1		SQQP3033-A- (2158)-SQS DP1 SQIT1013-B- (2145)-SQS DP3
Slot 7	SQIT3014-B- (2320)-DKG 6/5	SQQP3073-H- (2152)-SQS DP1	SQQP3063-B-(2966)/(2304)-DKG 6/5	SQIT1013-A- (2314)-SQS BT11	SQIT3033-B- (2307)-DKG 6/5

Slot number	Day 1 - Sunday	Day 2 - Monday	Day 3 - Tuesday	Day 4 – Wednesday	Day 5 – Thursday
	SQQP3033-B- (2158)-SQS DP1 SQQP4073-A- (2157)-SQS DP3	SQQP3073-C- (2190)-SQS DP3 SQQP3073-B- (2243)-SQS L4	SQQP3063-A-(2157)-SQS DP1 SQIT1013-C-(2313)-SQS DP3		
Slot 8	SQIT3014-B- (2320)-DKG 6/5 SQQP3033-B- (2158)-SQS DP1	SQQP2014-D- (1607)-DKG 6/5 SQQP3073-C- (2190)-SQS DP3 SQQP3073-B- (2243)-SQS L4	SQQP3063-B-(2966)/(2304)-DKG 6/5 SQIT1013-C-(2313)-SQS DP3	SQQP3073-D- (2243)-DKG 6/5 SQIT1013-A- (2314)-SQS BT11	SQIT3033-B- (2307)-DKG 6/5
Slot 9	SQIT3014-B- (2320)-DKG 6/5	SQQP2014-D- (1607)-DKG 6/5	SQIT1013-C-(2313)-SQS DP3	SQQP3073-D- (2243)-DKG 6/5 SQIT1013-A- (2314)-SQS BT11	
Slot 10		SQQP2014-D- (1607)-DKG 6/5 SQIT3014-D- (2311)-SQS DP3	SQIT3033-A-(2328)-SQS DP1 SQQP3073-F-(2321)-SQS DP3	SQQP3073-D- (2243)-DKG 6/5 SQIT1013-A- (2314)-SQS BT11	
Slot 11	SQQP3073-A- (2184)-DKG 6/5 SQQP3073-I- (3497)-SQS DP3	SQQP2014-D- (1607)-DKG 6/5 SQIT3014-D- (2311)-SQS DP3	SQQP3073-G-(3264)/ (737)-DKG 6/5 SQIT3033-A-(2328)-SQS DP1 SQQP3073-F-(2321)-SQS DP3	SQIT3014-A- (2321)-SQS DP1	SQIT3043-A- (2320)-DKG 6/5 SQQP2014-A- (5381)-SQS DP1
Slot 12	SQQP3073-A- (2184)-DKG 6/5 SQQP3073-I- (3497)-SQS DP3	SQIT3014-D- (2311)-SQS DP3	SQQP3073-G-(3264)/(737)-DKG 6/5	SQIT3014-A- (2321)-SQS DP1	SQIT3043-A- (2320)-DKG 6/5 SQQP2014-A- (5381)-SQS DP1

Slot number	Day 1 - Sunday	Day 2 - Monday	Day 3 - Tuesday	Day 4 – Wednesday	Day 5 – Thursday
Slot 13	SQQP3073-A- (2184)-DKG 6/5	SQQS3183-A- (1887)-DKG 6/5	SQIT3033-C-(2307)-DKG 6/5	SQIT3014-A- (2321)-SQS DP1	SQQP6023 -A- (2149)-DKG 6/5 SQQP2014-A- (5381)-SQS DP1
Slot 14		SQQS3183-A- (1887)-DKG 6/5	SQIT3033-C-(2307)-DKG 6/5	SQIT3014-A- (2321)-SQS DP1	SQQP6023 -A- (2149)-DKG 6/5 SQQP2014-A- (5381)-SQS DP1

From Table 4.3, there are no duplication or conflicts of constraints whereas giving 100% of feasible solution. Thus, this is the best solution obtained from GA in solving University timetabling problem.

4.3 Summary

This section successfully fulfilled the third objective, whereas to experiment the best parameter of GA model in solving University timetabling problem. According to the results of the model using the GA via Matlab software, it is concluded that the results of 27 simulations show that most of the percentage of solutions were close to 100% of the scheduled courses.

Based on the simulation result with the best fitness value at test run-17, all constraints were satisfied and timetable was generated with 100% solution. However, other test-run did not fully generate 100% solution and violate at least one (1) soft constraint.

The fitness function of the solution is negative because the fitness function is in the form of minimization, meaning that the more negative the value of the fitness function, the better it represents a solution that satisfies every constraint.

Based on comparison of the test run, the best solution obtained from test run-17 with the combination of parameter 500 for population size, 200 maximum iterations, 0.5 mutation rate and %5 elitism rate given the best fitness function which have reached -8871.5 fitness value. The combination of parameter setting at test run-17 able to provide a 100% solution or 41 of the 41 courses have been successfully assigned.

CHAPTER 5

CONCLUSION

5.1 Achievement of Research Objectives

In this thesis, the problem involved assignment for weekly periods of university timetabling to classroom in Decision Science Department. Adapted method was presented for solving timetabling problems of University courses based on GA. Overall, this thesis consists of five chapters. Chapter 1 explained the brief introduction of the research by referring to the problem statements, while Chapter 2 discussed the related literature review in solving university timetabling problem.

Subsequently, in Chapter 3, the research methodology is explained in terms of the necessary structures of University timetabling, formulation of the problem and conceptual model. The requirements of university timetabling problem that needs to be satisfied in generating feasible timetable has been identified at Section 3.4 as to fulfil the first objective. While the second objective which is to develop a GA model for solving university timetabling problem that satisfying constraints has been achieved at Section 3.6.

Then, Chapter 4 discussed on the result's analysis and the best parameter. In the proposed method, hard and soft constraints are used to determine feasible solutions with the best parameter setting. As predicted, using GA preferences for each lecturer are fully satisfied and can be used to solve real-world problem. The experimental result show that large population size lead to the best fitness value. An example of the best solution is displayed in a table which satisfy all hard and soft constraints. The final objective to solve the best parameter of GA in solving the University timetabling problem was achieved in Chapter 4.

As a final point, the research was concluded in this Chapter 5. Basically, this study managed to achieve the main purpose of this study that is to solve an application of university course timetabling problem by using genetic algorithm. All the research questions and objectives in this research have been successfully answered.

5.2 Contribution of the Study

The contribution of the study are as follows:

1. This study able to demonstrate the practicability of developed algorithm as an effective tool for obtaining optimal solutions in general university course timetable. Normally, longer time needed to generate timetable. By the application, the timetable can be generated in a short time. Hence, it is able to support the course coordinator to generate more feasible and effective timetable for every semester.
2. This study may also serve as a reference for future research on the similar topic of university timetabling problem where preferences on day and time, also assigning classroom based on number of students to be considered.

5.3 Limitation

This research may have several limitations due to some shortcomings in the test engine performance and time constraints. Besides, the user has to input each lecturer’s details in the Excel input data and the output is displayed in the Excel Output file. However, some combination of parameter cannot produce a feasible timetable.

5.4 Future work

Given the lack of benchmark data and evaluation methods for specific problems, and with no standard approach to evaluate the result, it is difficult to make a cross-comparison of the various algorithms.

Improving what has been done by emphasizing more on the requirements is one way to achieve this, with the hope of a better future. The ongoing research on this topic includes the solution of the mathematical model to generate a robust solution for course timetabling problem

For future work, the system can be further improved with user-friendly system interface where users able to add or delete the data without having to open the input files.

Besides, in the future work may investigate the list of soft constraints that can be revised and take into consideration such as assessing the workload of the lecturers who hold an administration cost and having more postgraduate students under their supervisions, and assigning lecturer to course based on their interest and area specialization.

After implementing the proposed algorithm, it was concluded that the approach showed a good performance in converging to the true optimal solution.

Moreover, the application can be further improved by hybridizing GA with other approaches or by modifying the function of GA operator.

References

Abdullah, S. (2006). Heuristic approaches for university timetabling problems (Doctoral dissertation, University of Nottingham).

Abdullah, S., Burke, E.K., & McColloum, B. (2005). An Investigation of Variable Neighborhood Search for University Course Timetabling. In The 2th multidisciplinary conference on scheduling: Theory and applications, NY, USA (pp. 413–427).

Abdullah, S., & Hamdan, A.R. (2008). A hybrid approach for university course timetabling. IJCSNS, 8(8), 127.

Abayomi-Alli, O., Abayomi-Alli, A., Misra, S., Damasevicius, R., & Maskeliunas, R. (2019). Automatic examination timetable scheduling using particle swarm optimization and local search algorithm. In Data, engineering and applications (pp. 119-130). Springer, Singapore.

Abramson, D., Amoorthy, M. K., & Dang, H. (1999). Simulated annealing cooling schedules for the school timetabling problem. Asia-Pacific Journal of Operational Research, 16(1), 1.

Adewumi, A. O., Sawyerr, B. A., & Montaz Ali, M. (2009). A heuristic solution to the university timetabling problem. Engineering Computations, 26(8), 972-984.

Akkan, C., & Gülcü, A. (2018). A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem. Computers & Operations Research, 90, 22-32.

Ahangaran, M., Pourbozorg, Talebi, M., & Soleymani, K (2017). Automatic Generation of University Course Timetabling Using Genetic Algorithm. 13th International Industrial Engineering Conference. Iran Institute of Industrial Engineering

Aladag, C. H., Hocaoglu, G. A., & Basaran, M. (2009). The effect of neighbourhood structures on tabu search algorithm in solving course timetabling problem. Expert Systems with Application, 36, 12349–12356.

AlHadid, I., Kaabneh, K., Tarawneh, H., & Alhroob, A. (2020). Investigation of simulated annealing components to solve the university course timetabling problem. Italian journal of pure and applied mathematics, 44, 291-301.

Alghamdi, H., Alsubait, T., Alhakami, H., & Baz, A. (2020). A review of optimization algorithms for university timetable scheduling. Engineering, Technology & Applied Science Research, 10(6), 6410-6417.

Al-Negheimish, S., Alnuhait, F., Albrahim, H., Al-Mogherah, S., Alrajhi, M., & Hosny, M. (2018). An intelligent bio-inspired algorithm for the faculty scheduling problem. International Journal of Advanced Computer Science and Applications, 9(5).

Alsmadi, O. M. K., Za'er, S., Abu-Al-Nadi, D. I., & Algsoon, A. (2011). A novel genetic algorithm technique for solving university course timetabling problems. In Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on (pp. 195-198). IEEE.

Alves, R. M., Cunha, F., Subramanian, A., & Brito, A. V. (2022). Minimizing energy consumption in a real-life classroom assignment problem. OR Spectrum, 1-27. https://doi.org/10.1007/s00291-022-00674-z

Al-Yakoob, S.M. and Sherali (2006), H.D. Mathematical Programming Models and Algorithms for a Class-Faculty Assignment Problem, European Journal of Operational Research, 173, pp. 488-507. 2006.

Andrew, G. M., & Collins, R. (1971). Matching Faculty to Courses. College and University.

Arratia-Martinez, N. M., Avila-Torres, P. A., & Trujillo-Reyes, J. C. (2021). Solving a University Course Timetabling Problem Based on AACSB Policies. Mathematics, 9(19), 2500.

Assi, M., Halawi, B., & Haraty, R. A. (2018). Genetic algorithm analysis using the graph coloring method for solving the university timetable problem. Procedia Computer Science, 126, 899-906.

Asmuni, H., Burke, E. K., & Garibaldi, J. M. (2005). Fuzzy multiple heuristic ordering for course timetabling. In The proceedings of the 5th United Kingdom workshop on computational intelligence (UKCI05), London, UK (pp. 302–309).

Aycan, E., & Ayav, T. (2008). Solving the course scheduling problem using simulated annealing. IEEE.

Aziz, N. L. A., & Aizam, N. A. H. (2018, September). A brief review on the features of university course timetabling problem. In AIP Conference Proceedings (Vol. 2016, No. 1, p. 020001). AIP Publishing LLC.

Babaei, H., Karimpour, J., & Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, *86*, 9. https://doi.org/10.1016/j.cie.2014.11.010

Babaei, H., Karimpour, J., & Hadidi, A. (2019). Generating an optimal timetabling for multi-departments common lecturers using hybrid fuzzy and clustering algorithms. Soft Computing, 23(13), 4735-4747.

Badri, M. A. (1996). A two-stage multiobjective scheduling model for faculty-course-time assignments. European Journal of Operational Research, 94(1), 16-28.

Bashab, A., Ibrahim, A. O., AbedElgabar, E. E., Ismail, M. A., Elsafi, A., Ahmed, A., & Abraham, A. (2020). A systematic mapping study on solving university timetabling problems using meta-heuristic algorithms. Neural Computing and Applications, 32(23), 17397-17432.

Berisha, A., Bytyçi, E., & Tershnjaku, A. (2017). Parallel Genetic Algorithms for University Scheduling Problem. International Journal of Electrical and Computer Engineering (IJECE), 7(2), 1096-1102.

Botangen, K. A. W., & Khan (2014), C. L. Class-Scheduling System for the Central Luzon State University.

Burke, E. K., Elliman, D., & Weare, R. (1994). A genetic algorithm based university timetabling system. In East-West Conference on Computer Technologies in Education, Crimea, Ukraine pp35-40.

Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. European Journal of Operational Research, 140(2), 266-280.

Carter, M. W., & Laporte, G. (1997). Recent developments in practical course timetabling. In International Conference on the Practice and Theory of Automated Timetabling (pp. 3-19). Springer, Berlin, Heidelberg.

Carter, M. W., & Tovey, C. A. (1992). When is the classroom assignment problem hard?. Operations Research, 40(1-supplement-1), S28-S39.

Chen, M., Tang, X., Song, T., Wu, C., Liu, S., & Peng, X. (2020). A Tabu search algorithm with controlled randomization for constructing feasible university course timetables. Computers & Operations Research, 123, 105007.

Chen, P. S., & Zeng, Z. Y. (2020). Developing two heuristic algorithms with metaheuristic algorithms to improve solutions of optimization problems with soft and hard constraints: An application to nurse rostering problems. Applied Soft Computing, 93, 106336.

Cheng, J. R., & Gen, M. (2019). Accelerating genetic algorithms with GPU computing: A selective overview. Computers & Industrial Engineering, 128, 514-525.

Cooper, T. B., & Kingston, J. H. (1995). The complexity of timetable construction problems. In International Conference on the Practice and Theory of Automated Timetabling (pp. 281-295). Springer, Berlin, Heidelberg.

Costa, D. (1994). A tabu search algorithm for computing an operational timetable. European Journal of Operational Research, 76(1), 98-110.

Davis, L. (1991). Handbook of genetic algorithms.

Daskalaki, S., Birbas, T., & Housos, E. (2004). An integer programming formulation for a case study in university timetabling. European journal of operational research, 153(1), 117-135.

Daskalaki, S., & Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. European Journal of Operational Research, 160(1), 106-120.

DavidWilson, I., Davies, R., & Stanton, N. (2013). A Genetic Algorithm based Solution to the Teaching Assignment Problem. *International Journal of Computer Applications*, *81*(19), 1-6.

Deris, S., Omatu, S., & Ohta, H. (2000). Timetable planning using the constraint based reasoning. Computers & Operations Research, 27, 819–840.

De Jong, K. A. (1975). Analysis of the behavior of a class of genetic adaptive systems.

Díaz-Parra, O., Fuentes-Penna, A., Barrera-Cámara, R. A., Trejo-Macotela, F. R., Fernández, J. C. R., Ruiz-Vanoye, J. A., ... & Rodríguez-Flores, J. (2022). Smart Education and future trends. Int. J. Comb. Optim. Probl. Informatics, 13(1), 65-74.

Dorneles, Á. P., de Araújo, O. C., & Buriol, L. S. (2017). A column generation approach to high school timetabling modeled as a multicommodity flow problem. European Journal of Operational Research, 256(3), 685-695.

Esteban, A., Zafra, A., & Romero, C. (2018). A Hybrid Multi-Criteria Approach Using a Genetic Algorithm for Recommending Courses to University Students. International educational data mining society.

Fen, H. S., Safaai, D., Hashim, M., & Zaiton, S. (2009). University course timetable planning using hybrid particle swarm optimization. *Conference on Intelligence and Human-Oriented Computing*, 93–99.

Garey, M. R. & Johnson, D. S. (1997). Computers and Intractability: A Guide to the theory of NP-Completeness. W. H. Freeman & Co.

Genc, B., & O'Sullivan, B. (2020, September). A Two-Phase constraint programming model for examination timetabling at university college cork. In International Conference on Principles and Practice of Constraint Programming (pp. 724-742). Springer, Cham.

Ghazali, N.H & Ramli, R. (2004). Past Solutions of Driver Scheduling and A Promising Path Via Genetic Algorithm. Seminar Kebangsaan Sains Pemutusan 2004, pp. 385 – 392.

Ghaffar A., Sattar, M. U., Munir, M., & Qureshi, Z. (2022). Multi-objective fuzzy-based adaptive memetic algorithm with hyper-heuristics to solve university course timetabling problem. EAI Endorsed Transactions on Scalable Information Systems, e14-e14.

Goh, S. L., Kendall, G., & Sabar, N. R. (2019). Simulated annealing with improved reheating and learning for the post enrolment course timetabling problem. Journal of the Operational Research Society, 70(6), 873-888.

Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning, 1989. Reading: Addison-Wesley.

Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. Foundations of genetic algorithms, 1, 69-93.

Gozali, A. A., & Fujimura, S. (2020). Solving University Course Timetabling Problem Using Multi-Depth Genetic Algorithm. In SHS Web of Conferences (Vol. 77). EDP Sciences.

Grefenstette, J. J. (1992, September). Genetic algorithms for changing environments. In PPSN (Vol. 2, pp. 137-144).

Gunawan, A., Ng, K. M., & Poh, K. L. (2007). Solving the teacher assignment-course scheduling problem by a hybrid algorithm. World Academy of Science, Engineering and Technology, 33, 259-264.

Gunawan, A., Ng, K. M., & Ong, H. L. (2008). A genetic algorithm for the teacher assignment problem for a university in Indonesia. Information and Management Sciences, 19(1), 1-16.

Gunawan, A., Ng, K. M., & Poh, K. L. (2008). A hybrid algorithm for the university course timetabling problem. Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling.

Gunawan, A., & Ng, K. M. (2011). Solving the teacher assignment problem by two metaheuristics. International Journal of Information and Management Sciences, 22(2), 73-86.

Gunawan, A., Ng, K. M., & Poh, K. L. (2012). A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem. Computers & Operations Research, 39(12), 3074-3088.

Gupta, S., & Sinha, S. (2020). Academic Staff planning, allocation and optimization using Genetic Algorithm under the framework of Fuzzy Goal Programming. Procedia Computer Science, 172, 900-905.

Hambali, A. M., Olasupo, Y. A., & Dalhatu, M. (2020). AUTOMATED UNIVERSITY LECTURE TIMETABLE USING HEURISTIC APPROACH. Nigerian Journal of Technology, 39(1), 1-14.

Hassanat, A., Almohammadi, K., Alkafaween, E. A., Abunawas, E., Hammouri, A., & Prasath, V. S. (2019). Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. Information, 10(12), 390.

Hossain, S. I., Akhand, M. A. H., Shuvo, M. I. R., Siddique, N., & Adeli, H. (2019). Optimization of university course scheduling problem using particle swarm optimization with selective search. Expert systems with applications, 127, 9-24.

Hosny, M., & Fatima, S. (2011). A survey of genetic algorithms for the university timetabling problem. International Proceedings of Computer Science and Information Technology, 13.

Hosny, M. I. (2012). A Heuristic Algorithm for Solving the Faculty Assignment Problem. In Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS) (p. 1). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Irene, S. F. H., Deris, S., & Zaiton, M. H. S. (2009, January). A study on PSO-based university course timetabling problem. In Advanced Computer Control, 2009. ICACC'09. International Conference on (pp. 648-651). IEEE. Chicago

Iwańkowicz, R. R., & Taraska, M. (2018). Self-classification of assembly database using evolutionary method. Assembly Automation.

Kazarlis, S., Petridis, V., & Fragkou, P. (2005). Solving university timetabling problems using advanced genetic algorithms. GAs, 2(7), 8-12.

Kirkpatrick S., Gelatt C. D. and Vecchi M. P. (1983), Optimization by Simulated Annealing, a publication of the American Association for the Advancement of Science, Vol. 220, No. 5498, pp. 671-680

Koksal, E., Hegde, A. R., Pandiarajan, H. P., & Veeravalli, B. (2021). Performance characterization of reinforcement learning-enabled evolutionary algorithms for integrated school bus routing and scheduling problem. International Journal of Cognitive Computing in Engineering, 2, 47-56.

Kusuma, P. D., & Adiputra, D. (2022). Lecturer-Course Assignment Model in National Joint Courses Program to Improve Education Quality and Lecturers' Time Preference. International Journal of Intelligent Engineering and Systems, 361-369.

Lambora, A., Gupta, K., & Chopra, K. (2019, February). Genetic algorithm-A literature review. In 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon) (pp. 380-384). IEEE.

Lamini, C., Benhlima, S., & Elbekri, A. (2018). Genetic algorithm based approach for autonomous mobile robot path planning. Procedia Computer Science, 127, 180-189.

Landa-Silva, D., & Obit, J. H. (2008). Great deluge with non-linear decay rate for solving course timetabling problems. In Intelligent Systems, 2008. IS'08. 4th International IEEE Conference (Vol. 1, pp. 8-11). IEEE. Chicago

Lindahl, M., Sørensen, M., & Stidsen, T. R. (2018). A fix-and-optimize matheuristic for university timetabling. Journal of Heuristics, 24(4), 645-665.

Mallicka, C. (2021). CLAPS: course and lecture assignment problem solver for educational institution using Hungarian method. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 3085-3092.

Marzouk, M., & Abdelakder, E. (2020). A hybrid fuzzy-optimization method for modeling construction emissions. Decision science letters, 9(1), 1-20.

Mayer, A., Nothegger, C., Chwatal, A., & Raidl, G. (2008). Solving the post enrolment course timetabling problem by ant colony optimization. In Proceedings of the 7th international conference on the practice and theory of automated timetabling.

Mitchell, M. (1998). An introduction to genetic algorithms. MIT press.

Maldonado-Matute, J. M., González Calle, M. J., & Celi Costa, R. M. (2020, February). Development of a solution model for timetabling problems through a binary integer linear programming approach. In International Conference on Intelligent Human Systems Integration (pp. 510-516). Springer, Cham.

Modibbo, U. M., Umar, I., Mijinyawa, M., & Hafisu, R. (2019). Genetic Algorithm for Solving University Timetabling Problem. Amity Journal of Computational Sciences (AJCS) 3 (1), 43, 50.

Mokhtari, M., Vaziri Sarashk, M., Asadpour, M., Saeidi, N., & Boyer, O. (2021). Developing a Model for the University Course Timetabling Problem: A Case Study. Complexity, 2021.

Mostafaie, T., Khiyabani, F. M., & Navimipour, N. J. (2020). A systematic study on meta-heuristic approaches for solving the graph coloring problem. Computers & Operations Research, 120, 104850.

Muklason, A., Bwananesia, P. C., YT, S. H., Angresti, N. D., & Supoyo, V. A. (2018, October). Automated Examination Timetabling Optimization Using Greedy-Late Acceptance-Hyperheuristic Algorithm. In 2018 International Conference on Electrical Engineering and Computer Science (ICECOS) (pp. 201-206). IEEE.

Muklason, A., Irianti, R. G., & Marom, A. (2019). Automated course timetabling optimization using tabu-variable neighborhood search based hyper-heuristic algorithm. Procedia Computer Science, 161, 656-664.

Muklason, A., Syahrani, G. B., & Marom, A. (2019). Great deluge based hyper-heuristics for solving real-world university examination timetabling problem: New data set and approach. Procedia computer science, 161, 647-655.

Niknamian, S. (2019). Proposing a Novel Mathematical Model and Meta-Heuristic Algorithm for University Course Timetabling with an Educational Quality Approach; a Case Study.

Nuntasen, N., & Innet, S. (2007). Application of genetic algorithm for solving university timetabling problems: A case study of Thai universities. UTCC Engineering Research Papers.

Ngo, S. T., Jaafar, J., Aziz, I. A., & Anh, B. N. (2021). A compromise programming for multi-objective task assignment problem. *Computers*, *10*(2), 1–16. https://doi.org/10.3390/computers10020015

Osman, I. H., & Laporte, G. (1996). Metaheuristics: A bibliography.

Puspitasari, F., & Moengin, P. (2020). Penerapan Metode Hybrid Genetic Algorithm (GA) dan Pattern Search (PS) untuk Penjadwalan Mata Kuliah Universitas. Jurnal Rekayasa Sistem Industri, 9(3), 201-212.

Qi, X., & Xu, L. (2012). The Application of Genetic Algorithm in Teaching Assignment Problem.

Razali, N. M., & Geraghty, J. (2011). Genetic algorithm performance with different selection strategies in solving TSP. In Proceedings of the world congress on engineering (Vol. 2, pp. 1134-1139).

Rezaeipanah, A., Abshirini, Z., & Zade, M. B. (2019). Solving University Course Timetabling Problem Using Parallel Genetic Algorithm.

Rezaeipanah, A., Matoori, S. S., & Ahmadi, G. (2021). A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search. Applied Intelligence, 51(1), 467-492.

Rjoub, A. (2020). Courses timetabling based on hill climbing algorithm. International Journal of Electrical and Computer Engineering (IJECE), 10(6), 6558-6573.

Saviniec, L., & Constantino, A. A. (2017). Effective local search algorithms for high school timetabling problems. Applied Soft Computing, 60, 363-373.

Saviniec, L., Santos, M. O., Costa, A. M., & dos Santos, L. M. (2020). Pattern-based models and a cooperative parallel metaheuristic for high school timetabling problems. European Journal of Operational Research, 280(3), 1064-1081.

Sastry, K., Goldberg, D. E., & Kendall, G. (2014). Genetic algorithms. In Search methodologies (pp. 93-117). Springer US.

Schniederjans, M. J., & Kim, G. C. (1987). A goal programming model to optimize departmental preference in course assignments. Computers & Operations Research, 14(2), 87-96.

Shatnawi, S., Al-Rababah, K., & Bani-Ismail, B. (2010). Applying a novel clustering technique based on FP-tree to university timetabling problem: A case study. IEEE.

Sultan, A. B. M. (2020). A genetic algorithm approach for timetabling problem: The time group strategy. Journal of Information and Communication Technology, 3(2), 1-14.

Tan, J. S., Goh, S. L., Kendall, G., & Sabar, N. R. (2021). A survey of the state-of-the-art of optimisation methodologies in school timetabling problems. Expert Systems with Applications, 165, 113943.

Turki Alotaibi, E., AyedAlonizi, E., Jeddoh, F. M., Montahaaliabalkhail, Algefari, S., Asheddy, A., & Kurdi , H. (2014). Solving Teacher Assignment Problem by Asynchronous Cooperative Parallel Genetic Algorithm. *International Journal of Information Technology & Computer Science (www.ijitcs.com) Volume 15 Issue No 1*, 48-69

Xu, M., & Zhou, J. (2020). Elite immune ant colony optimization-based task allocation for maximizing task execution efficiency in agricultural wireless sensor networks. Journal of Sensors, 2020.

Wang, B., Geng, Y., & Zhang, Z. (2019, October). Applying genetic algorithm to university classroom arrangement problem. In Journal of Physics: Conference Series (Vol. 1325, No. 1, p. 012157). IOP Publishing.

Wang, Y. Z. (2002). An application of genetic algorithm methods for teacher assignment problems. Expert Systems with Applications, 22(4), 295-302.

Wong, C. H., Goh, S. L., & Likoh, J. (2022, May). A Genetic Algorithm for the Real-world University Course Timetabling Problem. In 2022 IEEE 18th International Colloquium on Signal Processing & Applications (CSPA) (pp. 46-50). IEEE.

Yusoff, M., & Roslan, N. (2019, July). Evaluation of genetic algorithm and hybrid genetic Algorithm-Hill climbing with elitist for Lecturer University timetabling problem. In International Conference on Swarm Intelligence (pp. 363-373). Springer, Cham.