



Faculty of Computer Science and Information Technology

Computerised Heuristic Algorithm for Multi-Location Lecture Timetabling

Kuan Huiggy

**Master of Science
2020**

Computerised Heuristic Algorithm for Multi-Location Lecture Timetabling

Kuan Huiggy

A thesis submitted

In fulfillment of the requirements for the degree of Master of Science

(Computer Science)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK

2020

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

.....

Signature

Name : Kuan Huiggy

Matrik No : 16020143

Faculty : Faculty of Computer Science and Information Technology,
Universiti Malaysia Sarawak

Dated :

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude and appreciation to my supervisor, Dr. Sze San Nah for her valuable and constructive suggestions, guidance and encouragement during the phases of this research project. Besides that, great appreciation for her willingness to share and discuss her knowledge and insights on this case study which greatly assisted this research project. Apart from that, special thanks to Mr. Heng Chin Siong from Faculty of Cognitive Sciences and Human Development for his patience and time in implicitly providing the real dataset for this research project.

I also wish to extend my appreciation to Universiti Malaysia Sarawak for the Zamalah Vice Chancellor Award which provided me partial scholarship and made me pursuing my Master degree a reality. Many thanks to Faculty of Computer Science and Information Technology for providing a comprehensive learning environment which enabled me to focus on my research till completion.

I would also like to acknowledge with much appreciation for Mr. Lee Jun Choi, Dr. Cheah Wai Shiang, Dr. Chiew Kang Leng, Dr. Irwandi Hipiny, Dr. Stephanie Chua Hui Li and more in giving me a learning environment and supports throughout the completion of this research project.

ABSTRACT

This research focuses on multi-location coursework timetabling problem for Master of Science in Human Resource Development (MSc HRD) at the Faculty of Cognitive Sciences and Human Development (FCSHD), Universiti Malaysia Sarawak (UNIMAS). The MSc HRD degree is designed especially for working professionals to seek advanced knowledge, skills and confidence in the areas of human resource development and management. The courses are conducted during weekends. Apart from the main campus at Kota Samarahan it is also being offered at other learning centres in Malaysia, in order to fulfil the high market demand to obtain a master degree. Due to this, the lecturers are assigned to different teaching locations. This situation has made the timetabling of lectures very challenging. Moreover, the process of current manual timetabling practice to produce a clash-free timetable is time-consuming. Besides that, the current timetabling practice needs to fulfil constraints such as where lecturer's unavailable dates, different types of teaching slot, team-teaching among lecturers, the lecturer can only conduct one course or at one location at one time, total teaching hours for a course and even distribution of lecture sessions and lecturer duty. The objective of this study is to design a heuristic model for multi-location timetabling problem. A two-stage heuristic algorithm is proposed to solve the multi-location timetabling problem on MSc HRD coursework programme. The proposed two-stage heuristic algorithm consists of *Lecturer Grouping Stage* which allocates the lecturers into different team-teaching groups. After that, the algorithm proceeds to *Group Allocation Stage* in a round robin optimisation. The lecturer's unavailability is considered in Stage II as well. Real data from two semesters were collected from FCSHD to test the feasibility of the proposed solution. The simulator generates clash-free timetable in less than a minute, while fulfilling the unavailability dates

of lecturers and different types of teaching slot. On average, more than 80% of the timetabled days fall within the acceptable range of the week's break between lecture sessions. A set of sensitivity analysis has also been conducted under different scenarios, such as the unavailability dates of lecturers, teaching slot type for locations and team-teaching basis. The results show that the proposed solution is effective and robust in solving MSc HRD coursework programme multi-location timetabling problem.

Keywords: Course timetabling, lecturer scheduling, heuristics, two-stage heuristic, multi-location

Penjadualan Jadual Waktu Dengan Algorithma Heuristik

ABSTRAK

Kajian ini memberi tumpuan kepada penjadualan jadual waktu kursus bagi Sarjana Sains dalam Pembangunan Sumber Manusia (MSc HRD) di Fakulti Sains Kognitif dan Pembangunan Manusia (FSKPM), Universiti Malaysia Sarawak (UNIMAS). Program Sarjana Sains ini direka khasnya untuk profesional yang sedang bekerja supaya dapat belajar pengetahuan yang lebih luas, kemahiran dan keyakinan dalam bidang pembangunan dan pengurusan sumber manusia. Kursus-kursus program ini mengadakan kuliah selepas waktu kerja atau semasa hujung minggu sahaja. Program Sarjana Sains ini juga ditawarkan kepada beberapa institusi pembelajaran untuk memenuhi permintaan pasaran tinggi demi mendapatkan sijil sarjana sains di Malaysia, selain daripada kampus utama di Kota Samarahan. Oleh sebab itu, pensyarah-pensyarah di UNIMAS dikehendaki untuk mengajar sesi-sesi kuliah di institusi-institusi pembelajaran di Malaysia. Sehubungan dengan itu, situasi ini telah menyebabkan masalah perancangan dalam jadual masa kepada pensyarah untuk mengajar di pelbagai institusi. Selain itu, kaedah penjadualan semasa akan mengambil masa untuk menyelesaikan masalah pertembungan dalam jadual masa. Kekangan-kekangan dalam kajian ini perlu diselesaikan dengan kaedah penjadualan semasa seperti pensyarah yang tidak ketersediaan, jenis masa pembelajaran berdasarkan institusi dan peruntukan sesi kuliah yang seimbang. Objektif kajian ini ialah membentuk satu kaedah heuristik pengkomputeran untuk masalah penjadualan di pelbagai lokasi. Kaedah heuristik yang dicadangkan terdiri daripada dua peringkat untuk menyelesaikan masalah penjadualan jadual waktu dalam MSc HRD yang ditawarkan kepada pelbagai institusi. Kaedah tersebut melibatkan algorithma yang memperuntukkan pensyarah-pensyarah kepada beberapa kumpulan pembelajaran. Selepas

itu, peringkat kedua dari kaedah tersebut akan merancang jadual masa pensyarah dengan mengambil pertimbangan pensyarah yang tidak sedia ada. Dua semester dataset sebenar dikumpul dari FSKPM untuk menguji kaedah heuristik ini. Simulator ini boleh menghasilkan jadual waktu yang boleh dilaksanakan tanpa pertembungan dalam jangka masa yang pendek. Secara keseluruhannya, lebih daripada 80% jadual masa adalah dalam kurungan julat rehat minggu antara sesi kuliah yang boleh diterima. Sehubungan itu, analisis kepekaan telah dilaksanakan dengan pelbagai situasi seperti pensyarah yang tidak ketersediaan, jenis masa pembelajaran berdasarkan institusi dan kumpulan pembelajaran. Kesimpulannya, keputusan-keputusan simulator telah menunjukkan bahawa kaedah tersebut adalah berkesan dan sesuai untuk menyelesaikan masalah penjadualan jadual waktu FSKPM di pelbagai institusi.

Kata kunci: Penjadualan jadual waktu, perancangan pensyarah, kaedah heuristik dua peringkat, pelbagai institusi, lokasi

TABLE OF CONTENTS

	Page
DECLARATION	i
ACKNOWLEDGEMENT	ii
ABSTRACT	iii
<i>ABSTRAK</i>	v
LIST OF TABLES	xii
LIST OF FIGURES	xiii
LIST OF ALGORITHMS	xv
LIST OF ABBREVIATIONS	xvi
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	3
1.3 Research Objectives	4
1.4 Research Scope	4
1.5 Research Significance	4
1.6 Thesis Outline	5

CHAPTER 2: LITERATURE REVIEW	7
2.1 Introduction	7
2.2 University Course Timetabling	7
2.3 Existing Timetabling Algorithm	10
2.3.1 Graph Colouring	10
2.3.2 Heuristic	11
2.3.2.1 Multi-Stage Heuristic	11
2.3.2.2 Genetic Algorithm	12
2.3.3 Metaheuristic	13
2.3.4 Constraint Based Approach	14
2.3.5 Other methods	14
2.4 Multi-Location Timetabling Problem	15
2.5 Summary	18
CHAPTER 3: METHODOLOGY AND CASE STUDY	20
3.1 Introduction	20
3.2 Research Methodology	20
3.3 Case Study on FCSHD	22

3.4	Modelling and Formulation	27
3.4.1	Sets	28
3.4.2	Parameters	28
3.4.3	Constraints	29
3.5	Objective Function	29
3.6	Summary	29
CHAPTER 4: TWO-STAGE HEURISTIC ALGORITHM		31
4.1	Introduction	31
4.2	Data Pre-Processing	31
4.2.1	Total number of lecture sessions for a course.	32
4.2.2	Identify the critical lecturers	33
4.3	Two-Stage Heuristic Algorithm	34
4.4.1	Stage I: Lecturer Grouping Stage	35
4.4.2	Stage II: Group Allocation Stage	39
4.5	Summary	45
CHAPTER 5: RESULT AND ANALYSIS		46
5.1	Introduction	46

5.2	Dataset Description	46
5.3	Computational Analysis	47
5.3.1	Overall Solution	48
5.3.2	Computational Time	49
5.3.3	Two-Stage Heuristic Algorithm	50
5.3.4	Distribution of Teaching Schedule	52
5.3.5	Distribution of Workload on Team-Teaching	56
5.3.6	Unavailability dates of lecturer	59
5.4	Sensitivity analysis	62
5.4.1	No Team-Teaching	62
5.4.2	Same Teaching Type Slot	66
5.5	Summary	71
CHAPTER 6: CONCLUSION		74
6.1	Introduction	74
6.2	Conclusion of Research	74
6.3	Research Contributions	77
6.4	Future Work	77

LIST OF TABLES

		Page
Table 2.1	References on the multi-location timetabling problem.	17
Table 3.1	Teaching slot type based on multi-location.	25
Table 3.2	Types of the courses and the required teaching hours.	27
Table 4.1	Lecture sessions for a course.	33
Table 5.1	Dataset general information.	47
Table 5.2	Comparison of method.	48
Table 5.3	Computational time comparison on both datasets.	49
Table 5.4	Summary of workload on team-teaching.	56
Table 5.5	Group allocation comparison with unavailability dates of lecturer.	60
Table 5.6	Summary of lecturer unavailability and its category.	60
Table 5.7	Summary of workload on no team-teaching.	65
Table 5.8	Summary of workload on same teaching slot types.	68
Table 6.1	Objectives and achievements.	76

LIST OF FIGURES

	Page
Figure 3.1 Coursework program plan in multi-location.	23
Figure 4.1 Stages in the Two-Stage Heuristic Algorithm.	35
Figure 4.2 Lecturer Grouping Stage input output flow.	36
Figure 4.3 Group assignment based on teaching slot types.	37
Figure 4.4 Flowchart of the lecturer grouping stage.	39
Figure 4.5 Generating a matrix in Group Allocation Stage.	40
Figure 4.6 Process of the Group Allocation Stage with a list of lecturer groups.	42
Figure 4.7 Flow chart of lecturer unavailability checking.	44
Figure 4.8 Flowchart of group allocation stage.	44
Figure 4.9 Timeslot assignment for lecture sessions for each course.	45
Figure 5.1 Process of Stage I Lecturer Grouping Stage.	51
Figure 5.2 Process of Stage II Group Allocation Stage.	52
Figure 5.3 Teaching schedule comparison for dataset 2017/2018-1.	53
Figure 5.4 Teaching schedule break comparison for dataset 2017/2018-2.	54
Figure 5.5 Distribution of workload on dataset 2017/2018-1.	57

Figure 5.6	Distribution of workload on dataset 2017/2018-2.	57
Figure 5.7	Ideal distribution of teaching schedule on unavailability dates.	61
Figure 5.8	Teaching schedule comparison on no team-teaching.	63
Figure 5.9	Workload on no team-teaching on dataset 2017/2018-1.	64
Figure 5.10	Workload on no team-teaching on dataset 2017/2018-2.	65
Figure 5.11	Teaching schedule comparison on off-UNIMAS slot type.	67
Figure 5.12	Teaching schedule break comparison on UNIMAS slot type.	67
Figure 5.13	Distribution of workload on a one-day teaching slot type on dataset 2017/2018-1.	69
Figure 5.14	Distribution of workload on a one-day teaching slot type on dataset 2017/2018-2.	69
Figure 5.15	Distribution of workload on a consecutive two-day lectures on dataset 2017/2018-1.	70
Figure 5.16	Distribution of workload on a consecutive two-day lectures on dataset 2017/2018-2.	70

LIST OF ALGORITHMS

	Page
Algorithm 4.1 Lecturer Grouping Stage	38
Algorithm 4.2 Group Allocation Stage	42

LIST OF ABBREVIATIONS

CTTP	Course Timetabling Problem
ETTP	Examination Timetabling Problem
FCSHD	Faculty of Cognitive Sciences and Human Development
KL	Kuala Lumpur
LTTP	Lecture Timetabling Problem
MSc HRD	Master of Science in Human Resource Development
TITAN	Transparent Interactive Timetabling based on Agent Negotiation
UNIMAS	Universiti Malaysia Sarawak
UTTP	University Timetabling Problem

CHAPTER 1

INTRODUCTION

1.1 Introduction

Timetabling problem is defined as a task to assign a set of events to a limited number of timeslots and resources to satisfy the operational constraints in the problem (Burke & Petrovic, 2002; Lewis, 2008). Due to different operational constraints that are applied in each university timetabling problem, it is hard to formulate a general timetable solution to suit the needs of each university (Lewis, 2008). In general, the constraints in the timetable can be classified into two constraints: hard constraints and soft constraints. The hard constraints must be satisfied under any circumstances and have higher priority than the soft constraints. Therefore, the feasible solution can be made when all of the hard constraints in the problem are satisfied. Meanwhile, the soft constraints are those satisfaction is desirable but optional to fulfil. Thus, the soft constraints can be used to improve the quality of the feasible timetable solution.

Nowadays, automated university timetabling attracts the attention of researchers from the field of operational research and artificial intelligence since 1950s (Burke et al., 2001). A feasible timetable solution can be generated with manual solution by some heuristic with trial and error, but at the cost of effort and time (Almeida & Oliveira, 2015). However, the university timetabling problem usually consists of a larger set of operational constraints that are needed to satisfy. Subsequently, the university timetabling problem becomes complex and manual designed timetable method may unable to fulfil all operational constraints (Davoudzadeh & Rafeh, 2009). As the increasing number of students and the courses to be conducted, the automated timetabling able to provide a

feasible solution in a short time by using a computerised system. Therefore, in order to generate a feasible timetable solution for the practical use, most of the university timetabling problem have been simplified for the academic research purpose. Hence, scheduling a timetable is a great challenge to get a feasible timetable solution (Ferdoushi et al., 2013).

In the university timetabling problem, a larger set of operational constraints to be considered in the timetabling task makes the manual timetabling process becomes complicated (Petrovic & Burke, 2004). Thus, it is time-consuming and required a significant amount of effort when generating a set of feasible timetable solution (Chowdhary et al. 2014). The automated timetabling with different timetable solving methods can assisted in generating a feasible timetable solution while satisfying all the operational constraints. Over the decades, various effective timetabling methods have been introduced to solve the university timetabling problem. Despite of this, due to different requirements set by each university, the university adopts the different methods to meet their needs when generating the timetable solution.

In this research, the lecturers are required to travel in order to conduct the lecture sessions at multiple teaching locations. This research consists of similar requirements and operational constraints of the general university timetabling problem. Therefore, assignment to multi-location becomes a critical problem when travelling budget constraint is applied in this university timetabling problem. Due to lack of researches on the multi-location university timetabling problem, some researches refer to the similar timetabling problem such as nurse rostering problem. As such, the nurse rostering problem needs to assign the nurses to various timeslots and locations in order to carry out the duties (Han &

Kendall, 2003). Thus, the multi-location constraint can be considered as a research gap in the university timetabling problem (Meisels & Kaplansky, 2002; Han & Kendall, 2003).

1.2 Problem Statement

In this research, a case study of multi-location timetabling problem for Master of Science in Human Resource Development (MSc HRD) in Faculty of Cognitive Sciences and Human Development (FCSHD), Universiti Malaysia Sarawak (UNIMAS) has been studied.

Besides solving MSc HRD timetable with no clashing lectures and fulfil total teaching hours for the courses. MSc HRD also facing multi-location lecture timetabling, as the MSc HRD is being offered to different locations throughout. Due to the multi-location consideration, more manpower is involved. Therefore, team-teaching approach is being adopted and this has complicated the timetabling. In order to generate a clash-free timetable, various operational constraints such as lecturer's unavailability dates, different teaching slot-type, different locations, distribution of workload and travelling budget are considered in this case study. Currently, multiple timetable adjustments and revisions by the manual timetabling practice are carried out to solve the lecture's clashing issue.

The current timetable solution is not individual lecturer based schedule. Team-teaching basis is applied to the courses where two lecturers share the lecture sessions to conduct. Later of the timetabling, two lecturers may need to discuss and plan their individual timetable to conduct the lecture sessions for a course. This research implies that the multi-location and team-teaching constraint is crucial for MSc HRD coursework programme timetabling problem.

1.3 Research Objectives

The research objectives for this research are:

- (i) To design a heuristic model for multi-location coursework timetabling problem.
- (ii) To formulate a mathematical model of multi-location coursework timetabling problem.
- (iii) To conduct sensitivity analysis on the proposed heuristic for multi-location coursework timetabling problem.

1.4 Research Scope

The research scope for this research is defined as follow:

- (i) Limited to a case study on multi-location coursework timetabling problem for MSc HRD coursework programme at FCSHD, UNIMAS.
- (ii) Schedule lecture sessions for the courses, excluding the Research Paper course which are required to conduct only for a day throughout the semester and timetabling scheduling in mid-term and final examination.
- (iii) The application of this heuristic algorithm is for the course scheduling on lecture sessions at multi-location.
- (iv) The application is applied to UNIMAS lecturers only, not for part-timer.
- (v) Unavailability dates of lecturer, teaching slot type, team-teaching of lecturers, total teaching hours to be fulfilled are considered in this study.

1.5 Research Significance

The significance of this research is to develop an automated multi-location timetabling simulator for MSc HRD coursework programme in FCSHD to replace the

current timetabling practices. The aim for this research is to save time in generating a feasible and clash-free timetable solution. It can also make the rescheduling or timetable adjustments much easier when there are some changes in the data input or on the availability of lecturers. Besides that, the timetable simulator is able to tell the lecturers when and where to conduct the lecture sessions.

1.6 Thesis Outline

The following chapters are outlines as below:

Chapter 2: Literature Review, provides an understanding on the timetabling historical development. Timetabling problem and heuristic solving methods are discussed by categories. The research on multi-location timetabling problem are also discussed.

Chapter 3: Methodology and Case Study, explains the case study on the MSc HRD coursework programme multi-location timetabling problem in FCSHD in details. The hard and soft constraints that are defined in this research are listed. Variables and parameters are defined from the analysis of operational constraints. An objective function is set. Modelling and formulations of the problem are elaborated.

Chapter 4: Two-Stage Heuristic Algorithm, describes the stages in the proposed two-stage heuristic algorithm in detail. The data sets are pre-processed first before the *Lecturer Grouping Stage* and *Group Allocation Stage*. An overview of the heuristic algorithm is illustrated.

Chapter 5: Result and Analysis, analyses and compares timetable solutions by both the current timetabling practices and the proposed two-stage heuristic algorithm. Result analysis on distribution of lecture sessions and workload of lecturers are carried out on real

timetable solution and generated solutions. Unavailability dates of the lecturer are tested and evaluated for the quality of timetable solution. The last part of this chapter discusses the sensitivity analysis and its result.

Chapter 6: Conclusion, concludes the findings by completing this research in this chapter. The achievement of this research is presented. This chapter ended with some future plans that may improve the heuristic algorithm.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter discusses the review on the multi-location university course timetabling problem in order to investigate the methods and steps to generate a feasible timetable solution. The literature review is about reviewing previous work on the subject of university course timetabling. It forms the basis for further research in the area and fulfilling the necessary gaps. This review also focuses on various approaches and the requirements for solving the multi-location timetabling problem. This chapter begins with the discussion on the university course timetabling and algorithms to solve the problem.

2.2 University Course Timetabling

The timetabling problem can be categorised into school timetabling, university course timetabling and examination timetabling. In this case study, the university course timetabling problem is investigated because similar constraints of multi-location timetabling problem are defined. The university course timetabling deals with scheduling a set of lectures to different cohorts of students. This research fits the category of university course timetabling where the lectures are assigned to multi-location to conduct the lecture sessions.

The university course timetabling problem is proven a NP-hard where the required computation time to solve the problem is increased exponentially with problem size (Cooper & Kingston, 1995). A feasible timetable can be generated through manual solution by some heuristic with trial and error, but at the cost of effort and time (Almeida &

Oliveira, 2015). The university course timetabling problem often consists of a larger set of lectures for the students within a set of periods and venues to conduct the courses (Schaerf, 1999). According to Lewis (2008), each university has different set of operational constraints to be considered in the course timetabling problem before generating a feasible timetable solution. Due to the increasing number of students and courses, it is usually difficult to formulate a general timetabling algorithm for all of the universities.

According to Burke and Erben (2003), Padmini and Athre (2010) and Pandey and Sharma (2016), the university timetabling problem can be categorised into *university timetabling problem (UTTP)*, *course timetabling problem (CTTP)*, *lecture timetabling problem (LTTP)* and *examination timetabling problem (ETTP)*. Each category in the university timetabling problem has a different set of operational constraints to focus on in order to generate a feasible timetable solution. Thus, in the era of computerisation, the timetabling in university can be generated automatically by a computerised system. Hence, the automation of timetabling enables a quick and feasible solution while satisfying all the constraints in the university timetabling problem (McCollum, 2007). Besides that, the heuristic algorithm that is often adopted in the university course timetabling problem has gained the spotlight among the researchers in the past decade (Lewis, 2008) because of the ease of implementation in the algorithm to generate a feasible solution in a short time.

As noted by Lewis (2008), graph colouring based heuristic is associated with university timetabling problem in early algorithmic techniques. However, the graph colouring techniques can only adopt a simple set of operational constraints. This is due to the real-world problem in university timetabling that usually consists of a larger set of operational constraints. Thus, complex heuristic algorithms are then discovered and applied to different optimisation for the university timetabling problem in graph colouring

techniques (Lewis, 2008). Besides that, the time to generate a feasible timetable increases exponentially in graph colouring techniques due to the NP-hard nature in university timetabling problem (Daskalaki & Birbas, 2005; Socha et al., 2003).

As a result, many potential heuristic algorithms have been proposed and are able to show a remarkable result in some universities' timetables. There are a set of relatively successful applications to the actual university problems, such as Dammak et al. (2008), Ferdoushi et al. (2013), and Almeida et al. (2015). This is due to each university having a set of distinctive operational constraints in their university's course timetabling problem. Therefore, many researchers use different algorithms to solve the different goals of the university timetabling problem such as the feasible timetable solution to satisfy all the constraints or generate an optimised quality of the timetable or both. However, due to the NP-hard nature of the university timetabling problem, it is usually scaled down to a simplified problem in order for the purpose of research and algorithm comparison (Murray et al., 2006; Burke & Rudova, 2007). Therefore, it is difficult to satisfy all the operational constraints imposed by the problem without penalising the quality of the timetable.

Besides that, a feasible timetable solution do not always give a good quality timetable when the allocation of resources is the major concern in the university course timetabling problem. To analyse the benchmark of the proposed algorithms in getting a good quality timetable, the time resources of students or courses are studied for their academic performance (Fred, 2016). Fred (2016) mentions that a low quality timetable solution will cause lower academic performance of students due to poor time management issues. Besides that, as mentioned by Lindahl et al. (2018), rooms (in our case is multi-location) and teaching periods are dependent on each other and affect the quality of the timetable solution. Hence, a quality timetable solution will utilise the resources efficiently.

2.3 Existing Timetabling Algorithm

This research aims to deliver an algorithm to generate a feasible timetable solution for multi-location lectures in Universiti Malaysia Sarawak (UNIMAS). In order to analyse the case study of the multi-location timetabling problem, various algorithms literature are reviewed and discussed in this subsection. The techniques which are literature reviews are graph colouring, genetic algorithm, heuristic algorithm, linear programming and constraint-based approach.

2.3.1 Graph Colouring

Graph colouring is a simple and useful algorithm in a small university course timetabling (Lewis, 2008). Each data requirement acts as a vertex and the edges to connect to each possible data requirement in the graph colouring. The graph colouring algorithm uses a simple rule where no pair of vertices with a common edge is given the identical colour and minimises the usage of the colours in the algorithm (Lewis, 2008; Ganguli & Roy, 2017). The graph colouring can only be used when less operational constraints are applied in the university course timetabling problem. In spite of that, Lewis (2008) notes that many current timetabling algorithms use the fundamental heuristic information from graph colouring as the timetabling solution.

In order to solve a complex timetabling problem, graph colouring associates with another heuristic algorithms to satisfy a larger set of operational constraints. There are some examples of timetabling problem that have been solved by graph colouring method (Al-Omari & Sabri, 2006; Dandashi & Al-Mouhamed, 2010; Sabar et al., 2012) with smaller data sets. Despite that, the timetable solutions that are generated from graph

colouring do not meet the university requirement as the number of students and courses are increasing each year.

2.3.2 Heuristic

Heuristic gains more attention in the university timetabling problem research area because heuristic can give a quick and feasible timetable solution. The main aspect of the heuristic is the trial and error method to fulfil the hard constraints and soft constraints that are applied in the university's course timetabling problem (Lewis, 2008). As noted by Schimmelpfeng and Helber (2007), the heuristic can generate a feasible timetable solution which only require a short computational time. Heuristic approach is used on the specific problem which do not guarantee an optimal solution but usually generate a better timetable for practical use (Wahid & Hussin, 2016).

2.3.2.1 Multi-Stage Heuristic

A few research have been done on heuristic method with multi stage. The multi stages are implemented in heuristic algorithm because the stages can be used to obtain an initial feasible solution and then the solution obtained can be optimised in later stage. The multi-stage heuristic can reduce the difficulty and complexity of the problem. For example, Santiago-Mozos et al. (2005) uses subject assignment and evolutionary algorithm. Bong et al. (2015) uses two stages of heuristic algorithm in solving university course timetabling problem: Course Grouping and Timeslot Allocation. Dammak et al. (2008) uses Lecture Timetabling Heuristic and Course Timetabling Heuristic. Mushi (2011) proposed a two phase heuristic algorithm: Simulated Annealing and Steepest Descent to solve university course timetabling problem. A three-stage heuristic is proposed to solve the examination timetabling problem by Mauritsius (2018), the three stages consist of the Constructive

Heuristic, Kempe Chain Simulated Annealing, and the Multi Neighbourhood Simulated Annealing.

The heuristic algorithm that is employed in most of the universities only suits their course timetabling problem, thus, it is unsuitable for a common heuristic algorithm to be applied in other timetabling problems (Burke et al., 2006). Hence, the metaheuristic method can solve the general university course timetabling problem with a slight tuning of parameters to generate a better quality solution.

2.3.2.2 Genetic Algorithm

Genetic algorithm is a search algorithm based on natural selection and natural genetics mechanisms (Parera et al., 2016). According to Holland (1992), the genetic algorithm is introduced based on the natural evolution, which is a reproduction and mutation process. Genetic algorithm serves as a vector for the parameters to generate a set of potential solutions (Holland, 1992) before it undergoes a crossover and mutation process. Consequently, each set of potential solutions is measured by the fitness function values to search for an optimised solution (Sigl et al., 2003).

The stages in the genetic algorithm usually consist of chromosome representation, reproduction operator, crossover operator, mutation operator and fitness function evaluation (Almeida et al., 2015). In the stages of the genetic algorithm, the chromosome representation stage is where the data structures are chosen from the data collection phase. The next reproduction operator stage will use the data structures from the previous stage and generate a set of potential timetables. Meanwhile the stage of crossover operator and mutation operator undergoes the selection process and correction process to produce a

feasible timetable. At the end of the genetic algorithm, the values from the fitness function evaluates the quality of the generated timetable.

Genetic algorithm has been discussed and used by Adewumi et al. (2009), Pillay (2016), Chaudhuri & De (2010) and Thanh (2007). The main drawback from this genetic algorithm is the researchers have to input the number of loops to generate the timetable solution and fitness function value to determine the quality of the solution.

2.3.3 Metaheuristic

Metaheuristic method is a problem independent method that can be applied to the wide range of the university's course timetabling problem. It uses other approaches such as simulated annealing, genetic algorithms, tabu search and other methods to produce a higher quality of timetable solutions (Burke & Newall, 2004). Generally, the metaheuristic method uses one or more initial solutions associated with the search algorithm for an optimised result (Burke & Petrovice, 2002).

From the survey of Lewis (2008), most of the metaheuristic algorithm can be categories into: *one-stage optimisation algorithms*, *two-stage optimisation algorithms* and *algorithms that allow relaxations*. One-stage optimisation algorithms allow both hard and soft constraints to be penalised while finding a timetable solution that adequately satisfies the constraints (Schearf, 1996; Lewis, 2008). Meanwhile, the two-stage optimisation algorithms are where the feasible solution is found in the first stage and soft constraints are then evaluated for optimisation in the second stage (Burke et al., 2003; Lewis, 2008). Algorithms that allow relaxations are applied where the constraints are relaxed to achieve a feasible timetable solution while satisfying both the constraints (Burke et al., 2001; Lewis, 2008).

An example of the metaheuristic method can be found from the electromagnetic-like mechanism (EM) and the great deluge algorithm (GD) by Abdullah et al. (2010), Ant Colony Optimisation (Socha et al., 2002), Memetic algorithm (Rossi-Doria & Paechter, 2004), Simulated Annealing (Kostuch, 2004), Tabu Search (Aladag & Hocaoglu, 2007) and Universiti Malaysia Sabah Labuan International Campus (Junn et al., 2017b).

2.3.4 Constraint Based Approach

Rescheduling of timetable solution is always needed when there is insufficient manpower resources in the university's course timetabling problem. In the process of rescheduling, a feasible timetable solution needs a rework and costs a lot of time to find a new solution to satisfy all the constraints. A constraint-based approach is used to find a new feasible solution for the replacement in the shortest time possible. For example, Kaewchanid and Wangmaeteekul (2016) have proposed the constraint-based approach to solve the timetabling problem in allocating teaching assistants. Woumans (2016) has used a spreading costs to the constraints in order to distribute the exam periods evenly for students who have examination timetabling problem. Junn et al. (2017a) uses a constraint-based approach to fulfil the hard constraints but not the soft constraints in order to generate a feasible solution in university's course timetabling problem.

2.3.5 Other methods

Besides that, there are other approaches to solve the university's course timetabling problem which include Constraint Logic Programming (Goltz & Matzke, 1999), Scheduling Agents (Meisels & Kaplansky, 2003), Hyper-heuristic (Han & Kendall, 2003) and Multi-Agent System (Houhamdi et al., 2019). These alternative methods can solve

higher levels of interactions between the operational constraints and generate an optimised timetable solution.

2.4 Multi-Location Timetabling Problem

This research focuses on the multi-location timetabling problem where the lectures are conducted in different teaching locations. This section reviews the researches who have proposed various methods to solve the multi-location timetabling problem.

Goltz and Matzke (1999) have designed the Constraint Logic Programming to cater to the lectures that are assigned at two different locations in Berlin. The distance between the two locations are considered in the hard constraints while generating a feasible timetable solution for the university timetabling problem. However, the solution involves a lot of manual work from the user interaction to remove any contradictions that are generated in the timetable solution (Goltz & Matzke, 1999).

Besides that, the Scheduling Agents method is proposed by Meisels and Kaplansky (2002) in order to solve the travelling cost constraint of teaching for multi-location in the university timetabling problem. Meisels and Kaplansky (2002) refer to the case study where the nurses are required to travel to different locations to carry out their duty. In the Scheduling Agents method, a Central Agent is used to interact with the Scheduling Agents to get a set of feasible timetable solution. A Central Agent acts as an entity to search for a feasible solution from the Scheduling Agents by searching for the combination where the constraints are satisfied. The problem is the number of iterations to run the Scheduling Agents method increases exponentially as the size of the search space grows in finding an optimal solution.

Additionally, Han and Kendall (2003a) adopts the Guided Operators for a Hyper-Heuristic Genetic Algorithm in the distribution of trainers to multi-location. A total of 10 training locations with 25 trainers is used in their case study. They improve the genetic algorithm method by removing the bad solution and optimising the solution with a good solution from the search space. The results show a remarkable quality timetable solution by iterating to find a good combination of feasible solutions. However, a higher cost of computational time is caused in finding a suitable combination with the genetic algorithm method.

In a previous work by Han and Kendall (2003b), the improved algorithm called Tabu Assisted Hyper-Heuristic Genetic Algorithm is proposed to solve the distribution of trainers to multi-locations. The objective of this method is to prioritise the courses and minimise the total travel penalty for the trainers with the assistance of the Tabu Method (Han & Kendall, 2003b). The Tabu method is used to improve the search space for a feasible solution while reducing the computational time as caused by the genetic algorithm method. As oppose to previous work, the computational time is costly because of the iteration steps in removing the bad solution and finding another feasible solution in the genetic algorithm method. This algorithm shows that an improved quality timetable solution can be generated in a short time.

In the next study, Adriaen et al. (2003) has proposed the Transparent Interactive Timetabling based on Agent Negotiation (TITAN) for the university's course timetabling problem. In this case study, four departments are shared with the six campuses for the lecture sessions. Thus, the venues for course teaching are shared by four departments. TITAN uses the metaheuristic algorithm and agent-based negotiation to generate a set of quality timetable solutions.

Sarin et al. (2010) uses Bender's partitioning to minimise the travel distances and the time of travel from their offices to the classrooms by faculty members. In this case study, the number of available classrooms is limited for each department building. The faculty members need to travel to the available classrooms in other department buildings to conduct the courses. In another example by Mirrazavi et al. (2003), a two-phase multiple objective approach is used to solve the course timetabling problem at different university sites. The travelling time is considered between the sites for students and lecturers. The two phases approach used are Room Allocation and Time Allocation with genetic algorithm. Di Stefano and Tettamanzi (2001) uses evolutionary algorithm to solve school timetabling problem. In this case study, the locations to conduct are one of the soft constraints which teachers prefer time of travel between the locations to be minimised in order to concentrate on their lessons.

In order to solve the multi-location timetabling problem in UNIMAS, a set of related papers are studied to study the comparison on the methods used. The studies show that the proposed methods are able to satisfy the multi-location teaching constraint on the timetabling problem and generate a feasible solution. To summarise the section of multi-location timetabling problem, Table 2.1 shows a summary on the finding of the case study in multi-location timetabling problem.

Table 2.1: References on the multi-location timetabling problem.

Author	Method	Problem Description
Goltz and Matzke (1999)	Constraint Logic Programming	Lecture assigning at two locations.

Table 2.1 continued

Di Stefano and Tettamanzi (2001)	Evolutionary algorithm	Time of travel between the locations to be minimised in order to concentrate on lessons.
Meisels and Kaplansky (2002)	Scheduling Agents	Travelling cost of the transports in wards.
Han and Kendall (2003a)	Guided Operators for a Hyper-Heuristic Genetic Algorithm	Prioritise the courses and minimise the total travel penalty for the trainers
Han and Kendall (2003b)	Tabu Assisted Hyper-Heuristic Genetic Algorithm	Improve previous work, reduce computational time.
Adriaen et al. (2003)	Transparent Interactive Timetabling based on Agent Negotiation (TITAN)	Four departments are shared with the six campuses for the lecture sessions
Mirrazavi et al. (2003)	Two-phase multiple objective approach	Consider travelling time between the sites for students and lecturers.
Sarin et al. (2010)	Bender's partitioning	Minimise the travelling distance and time of travel from their offices to classrooms by the faculty members.

2.5 Summary

Based on the previous literature review, few case studies have been conducted on the multi-location university timetabling problem. Although there are successful

applications on the heuristic method in most of the research, the real world problem usually consists of complex constraints with the increasing number of data sets per year. Thus, most of the heuristic methods in the university timetabling problem use a simplified pre-defined problem in order to generate the timetable solution in a short time. Hence, in our case study, there is a research gap for the multi-location university's course timetabling problem. As this research focuses on the multi-location timetabling problem with the similar operational constraints applied in most of the universities, the heuristic method is adopted as it is simple and direct to generate a feasible timetable solution. The heuristic algorithm is designed with *Lecturer Grouping Stage* and *Group Allocation Stage* with solution optimisation. Therefore, a two-stage heuristic algorithm is developed in this research due to difficulty and complexity of multi-location lecture timetabling problem.

CHAPTER 3

METHODOLOGY AND CASE STUDY

3.1 Introduction

This chapter discusses the methodology and case study of multi-location lecture timetabling of the MSc HRD at UNIMAS. This chapter presents the identified issues that analyse its operational constraints and formulate the multi-location timetabling problem. This chapter begins with a brief discussion on the methodology that is presented in this research.

3.2 Research Methodology

The flows of the research methodology in this research are as follow:

Step 1: Data collection and analysis

- Interview session with MSc HRD timetable planner (coordinator) in FCSHD are conducted.
- A case study on multi-location timetabling problem is studied.
- Real data sets are collected.
- Real timetable solution from MSc HRD coursework programme are analysed.

Step 2: Problem formulation

- Hard and soft operational constraints are defined.
- Variables and parameters are defined.
- Objective function is defined.
- Modelling and formulation are defined.

Step 3: Algorithm development

- Data pre-processing algorithm is developed to validate and verify the data sets for main algorithm.
- A two-stage heuristic algorithm is proposed to solve multi-location timetabling problem, namely *Lecturer Grouping Stage* and *Group Allocation Stage*.

Step 4: Testing and Evaluation

- Two real data sets that collected from earlier step is evaluated with the proposed two-stage heuristic algorithm to test its validity.
- Computational analysis has been done to check the performance of the two-stage heuristic algorithm.
- The real timetable solution and generated solutions are compared.
- Result analysis on distribution of lecture sessions and workload of lecturers have been done on the timetable solution.
- Unavailability dates of lecturers are tested and evaluate the quality of timetable solution.

Step 5: Sensitivity analysis

- Sensitivity analysis is carried out by adjusting the parameters and problem instance to test the robustness and quality of the proposed heuristic algorithm.
- The cases that are carried out are teaching slot types and no team-teaching basis.

Step 6: Result Compilation

- Thesis on full report of case study, methodology, algorithm and result.

3.3 Case Study on FCSHD

FCSHD offers MSc HRD by coursework to cater to the need of working adults to upgrade themselves. This master programme is conducted at multi-location such UNIMAS main campus, Sibul learning centre, Kuala Lumpur learning centre and so on. The programme takes a minimum of two years (or four academic semesters) duration to complete in order to obtain a master degree in HRD. Due to the increasing intake in the number of the students and the teaching locations, a total of 20 teaching staff in HRD by 2018 are participating in conducting the MSc HRD coursework program plan.

Currently, the timetable is planned manually before the start of new semester. The timetable solution goes through multiple revisions and adjustments in order to satisfy all the operational constraints and to avoid timetable clashes. There are several key issues with the current multi-location timetabling problem in MSc HRD of FCSHD:

(a) Multi-Location

This timetabling problem focuses on scheduling of courses to lecturers at the multi-location. Currently, the master coursework programme plan introduces the course teaching at multi-location besides Kota Samarahan which is known as off-UNIMAS campus. The off-UNIMAS campuses that are expanded for MSc HRD coursework programme plan are Kuala Lumpur, Sibul, Kota Kinabalu, Ipoh and Miri. There are a total of six teaching locations to be conducted and the UNIMAS lecturers need to travel to these locations to conduct the lecture sessions for the courses. Starting from September 2017, MSc HRD has officially offered at off-UNIMAS campuses, such as Kuala Lumpur (KL) and Sibul as illustrated in Figure 3.1 due to the high market demand of Human Resource Development.

Subsequently, the students are able to further their postgraduate study without travelling to the UNIMAS main campus which is situated at Kota Samarahan, Sarawak.

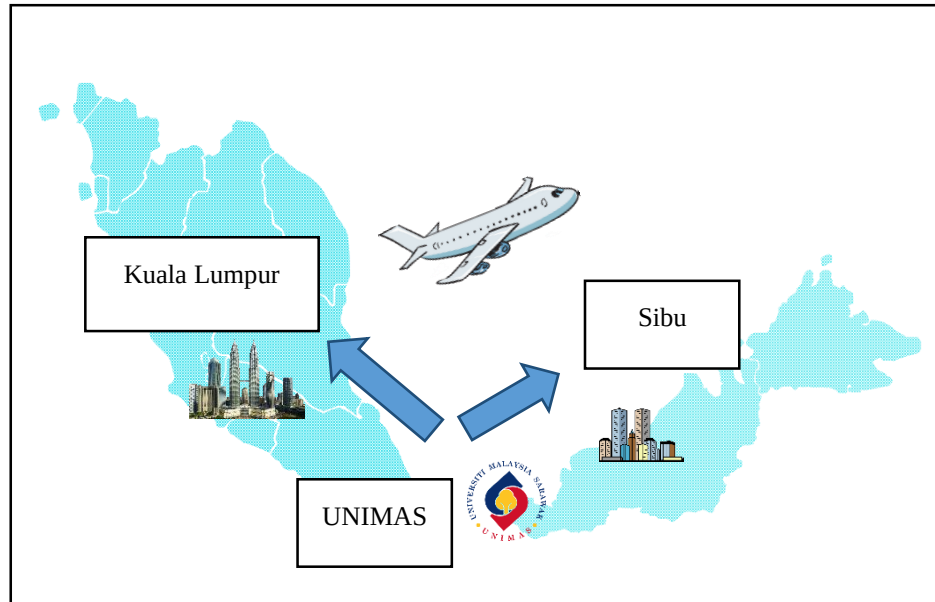


Figure 3.1: Coursework program plan in multi-location.

(b) Team-teaching

Same group of academic are assigned to teach all the courses at multi-location. Limited manpower resources and more courses to be conducted at multi-location cause difficulty in getting a feasible timetable solution. The current timetabling practices allow team-teaching of lecturers. This arrangement allows a second lecturer to back up the course teaching while another is busy with other commitments and travelling demand. For example, the second lecturer can help in conducting the lecture sessions when the main lecturer is out of duty or is unavailable for a week.

(c) Type of Schedule

The real timetable solution that are generated from current timetabling practices focuses on the course-based schedule. Currently, the current timetabling practices allocate the timeslots for the courses for each student-cohort without clashes. The team-teaching arrangement is applied in timetable solution to relax the constraints of limited manpower resources. However, from the analysis of the real timetable solution, individual assignment of lecture sessions for each lecturer is not considered. The individual assignment for each lecturer is time-consuming and multiple adjustments are required in order to generate a feasible timetable solution. Thus, each course teaching team have to discuss to plan internally for individual timetable, in order to fulfil the total teaching hours for the course. Hence, the timetable solution only optimises for the performance of students but not the lecturers. Therefore, the optimisation of the timetable solution needs to focus on both the availability of lecturers and the timetable of students.

(d) Student Cohorts

Cohort is a group of students who shares the similar courses plan together. A student can only participate in one particular study cohort throughout academic plan. Throughout the two years or four semesters of the master coursework programme, there are a total of four different student cohorts that can participate in one teaching location. Therefore, MSc HRD has offered a different set of courses for each student-cohort. Hence, a student can only be involved in one cohort at one time.

(e) Timeslots and Teaching Slot Types

All lecture sessions are conducted during the weekends only, namely, Saturday and Sunday. As in UNIMAS main campus, each weekend has two teaching slots: morning session which is from 9.00 a.m. to 12.00 p.m. and the afternoon session which is from 2.00 p.m. to 5.00 p.m. Meanwhile, the off-UNIMAS campuses such as in Kuala Lumpur and Miri, there is only one teaching slot which is from 9.00 a.m. to 5.00 p.m. with an hour break in between.

Table 3.1: Teaching slot type based on multi-location.

Location	UNIMAS		Off-UNIMAS campuses
Teaching Slot	Saturday	Sunday	Saturday and Sunday
Morning	9.00 a.m. to 12.00 p.m.	9.00 a.m. to 12.00 p.m.	9.00 a.m. to 5.00 p.m.
Afternoon	2.00 p.m. to 5.00 p.m.	2.00 p.m. to 5.00 p.m.	
Total hours for a lecture session in a week	6	6	14
Weeks to complete	7	7	3
Break	12.00 p.m. to 2.00 p.m.		One hour break

Each lecture session at UNIMAS main campus is required to be conducted on a one-day basis on either Saturday or Sunday, where two teaching slots (a total of six hours) are conducted. Meanwhile, a lecturer is required to conduct a session of consecutive two-day classes on both Saturday and Sunday at off-UNIMAS campuses. The main reason for consecutive two-day classes is due to a travelling budget. Consequently, a lecturer can only teach one course at one time or at one weekend, and one location at a time. Table 3.1 shows the teaching slot type and lecture session for the coursework program plan based on multi-location.

(f) Academic Calendar

Both UNIMAS and off-UNIMAS campuses comply 14 academic weeks of lectures in a semester. One week of semester break is allocated in the mid-semester academic calendar and no classes are conducted, for example in week 7. The academic semester calendar starts on the same week for all teaching locations. The exam week is allocated at the end of semester, e.g. week 15. Besides that, no classes are allowed during public holiday.

(g) Unavailability dates of lecturers

A lecturer might not be available for some weeks. The current timetabling practices require more adjustments to accommodate the lecturer unavailability constraint. The lecturer unavailability constraint has increased the difficulty of the timetable planning where the revision ought to fulfil the total of 42 teaching hours for each course as well.

(h) Courses

The MSc HRD coursework programme plan consists of coursework courses and research paper. The courses are categories as *faculty core courses* which are compulsory for the students to attend; meanwhile, *elective courses* which are offered according to the student's interest and lecturer's preference. Each core and elective course require a total of 42 teaching hours to conduct, and must be fulfilled before an end-semester exam is carried out for the semester. Besides that, *specialisation courses* such as Research Paper 1 and Research Paper 2 are conducted for the final year postgraduate students. These courses require one day lecture session to be carried out for the semester. The students are expected to complete the research paper as a partial requirement in order to graduate. Table 3.2 summarised all types of courses that are offered by FCSHD.

Table 3.2: Types of the courses and the required teaching hours.

Course Type	Required Teaching Hours	No. of Courses	Required lecture days	
			UNIMAS	Off-UNIMAS
Core course	42	11	7	6
Elective course	42		7	6
Specialisation course/Research Paper	6	2	1	1

3.4 Modelling and Formulation

In this multi-location timetabling problem, the following notation is used in the mathematical modelling.

3.4.1 Sets

C : set of courses = $\{c_1, c_2, \dots, c_n\}$

S : set of cohorts of students corresponding to locations = $\{s_{11}, s_{12}, \dots, s_{pq}\}$

L : set of lecturers = $\{l_1, l_2, \dots, l_m\}$

T : $T_l = \{T_l^{campus}, T_l^{off-campus}\}$ with set of teaching slot types = $\{t_1, t_2, \dots, t_r\}$

V : set of locations = $\{v_1, v_2, \dots, v_p\}$

E_{c_i} : set of lecturers of courses $c_i l_m l_b$

Set C is a set of courses to be conducted by a set of lecturers L in a set of locations V . Set S is a set of cohorts of students take the similar curriculum of courses in each location. Each course $c_i l_m l_b$ has a set of individual or team-teaching lecturers to conduct the lecture sessions which is defined as E_{c_i} . There is an assisting lecturers l_b to team-teach for course c_i , else $c_i l_m$ if no team-teaching for the course.

3.4.2 Parameters

n = number of courses

m = number of lecturers

r = number of lecture sessions for a course

p = number of locations

q = cohort of students

3.4.3 Constraints

From the case study as described above, the operational constraints can be categorised into a set of hard constraints (HC) and soft constraints (SC). Hard constraints are those satisfaction which is a must to fulfil to get a feasible solution while soft constraints are optional but assure a better-quality solution.

- HC₁ A lecturer can only conduct a course at one time.
- HC₂ A lecturer can only conduct at a location at one time.
- HC₃ A student can only attend one course at one time.
- HC₄ A student can only be involved in one cohort at one time.
- HC₅ Total teaching hours for a course must be fulfilled, i.e. 42 hours.
- HC₆ Fulfil the unavailability dates of lecturers.
- HC₇ Fulfil the teaching slot type based on the location, i.e. one course per week (off-UNIMAS campus) or two courses per week (UNIMAS campus).
- SC₁ Even distribution of all courses.
- SC₂ Even distribution of all lecture duty.

3.5 Objective Function

To minimise the lecture session gap of the courses for the students.

3.6 Summary

The case study on the real-life multi-location timetabling on the MSC HRD coursework program in FCSHD has been discussed. All set, parameters, objective function and operational constraints are well defined. As of this case study, timetabling becomes

more complex than normal lecture timetabling when multi-location consideration is involved in generating a feasible timetable solution.

The current timetabling practice is able to generate a clash-free timetable by manual and trial and error. However, the timetabling practice is time-consuming when team-teaching based is considered in timetabling. In order to satisfy all the operational constraints, multiple adjustments and revisions of timetable are carried out to solve the team-teaching clashing issue.

The operational constraints such as lecturer's unavailability dates and distribution of workload are not considered by the current timetabling practice. Therefore, this research has proposed a two-stage heuristic algorithm to solve the multi-location lecture timetabling problem. The next chapter will discuss the proposed two-stage heuristic algorithm to solve the multi-location timetabling problem.

CHAPTER 4

TWO-STAGE HEURISTIC ALGORITHM

4.1 Introduction

This chapter discusses the proposed two-stage heuristic algorithm in multi-location timetabling problem. The diagrams or illustrations and the algorithms are included to describe the stages in two-stage heuristic algorithm.

The heuristic algorithm is proposed because of its simplicity to cope with abundant constraints (Lewis, 2008). As noted by Schimmelpfeng and Helber (2007), the heuristic algorithm is able to generate a feasible solution in a short computational time. Heuristic approach is used on the specific problem which do not guarantee an optimal solution but usually generate a better solution for practical use (Wahid & Hussin, 2016). Therefore, it can be observed that the heuristic algorithm approach is widely applied in the real life including university timetabling problem.

The multi-location timetabling problem is solved with two stages namely *Lecturer Grouping Stage* and *Group Allocation Stage*. The two stages are applied in this heuristic algorithm because each stage can fulfil a set of constraints and optimise the allocation method. Before the real data sets are processed with the two-stage heuristic algorithm, a data pre-processing is conducted and discussed in next section.

4.2 Data Pre-Processing

In this section, a data pre-processing is conducted prior to the two-stage heuristic algorithm. The data inputs required are teaching locations, teaching courses respectively to student cohorts, assigned lecturers, teaching slot types based on locations and the

unavailability dates of lecturers. This is to ensure the correctness of data for this research and heuristic modelling. There are a few considerations to look into during the data pre-processing process and are identified as follow:

- (1) Total number of lecture sessions for a course.
- (2) Identify the critical lecturers.

4.2.1 Total number of lecture sessions for a course.

In UNIMAS main campus, a six-hours per lecture session, $T_l^{campus} = 6$ for a course is conducted on either Saturday or Sunday. Meanwhile off-UNIMAS campuses have a 14-hours, $T_l^{off-campus} = 14$ (consecutive two-day lecture) lecture requirement for each course. Thus, two courses can be conducted in the same week in UNIMAS campus; whereby off-UNIMAS campus can only conduct one course per weekend. Hence, courses that are conducted in UNIMAS main campus take seven weeks (seven lecture sessions in this case) to complete. Meanwhile, it takes three weeks (three lecture sessions or three full weekends) to complete a course at an off-UNIMAS campuses. Thus, the week, T_w that is taken for the lecture sessions for a course can be calculated from equation 4.1. Table 4.1 shows the total weeks to complete a course and its total number of lecture sessions.

$$T_w = T_t / T_l \quad \text{Equation 4.1}$$

Besides that, the lecturers are assigned to multi-location to conduct the lecture sessions for a course. Usually an academic week consists of 14 weeks for a semester, which consists of 12 hours per week to conduct the courses. In order to fulfil the 42 total teaching hours, T_t for a course, the maximum number of courses that can be conducted is four per cohort, which is derived from the equation 4.2. This also means off-UNIMAS campuses only

allow one lecture session for a week, thus each week will represent a group of lecturers to conduct the lecture sessions at multi-location. Currently, the lecturers are on team-teaching (pair-teaching) basis to conduct the lecture sessions for a course. Two lecturers share an equivalent amount of lecture sessions for a course in order to fulfil the total teaching hour for the course. In this case, the total teaching hours required is 42 hours. Therefore, this team-teaching basis allows the second lecturer to conduct a lecture session in other location while the main lecturer is out of duty. Hence, a lecturer may conduct maximum of nine different courses (on UNIMAS slot type) at multi-location in a semester by team-teaching by 18 hours for a course, where 12 hours are allocated for each week.

$$max_group = T_l * 14 weeks / T_t \quad \text{Equation 4.2}$$

Table 4.1: Lecture sessions for a course.

Location	UNIMAS		Off-UNIMAS campuses
Teaching Slot	Saturday	Sunday	Saturday and Sunday
Hours in a lecture session	6	6	14
Total weeks for a course	7	7	3
Total teaching hours for a course			42

4.2.2 Identify the critical lecturers

The lecturer with more teaching workloads is identified as the critical lecturer. In this timetable generation, the “criticalness” is identified as workload by the number of

teaching courses to conduct by each lecturer. Due to limited number of lecturers, a lecturer may conduct more than one course at different teaching locations. The analysis on the “criticalness” of critical lecturers helps to make a better planning for the timetabling. Thus, a proper planning is required to avoid the clash of lecturers and courses in the timetable and assure a feasible solution for the manpower resources. There are a few steps in determining the critical lecturers:

Step 1: List all the lecturers.

A set of lecturers is defined as $L = \{l_1, l_2, \dots, l_m\}$. Each lecturer has a total number of teaching courses. Based on the equation (4.2), a lecturer cannot conduct more than eight courses in one semester on the basis of team-teaching. This is to make sure the teaching courses are fairly distributed to each lecturer without over stressing one.

Step 2: Identify the criticalness.

In this second step, all the lecturers are sorted accordingly to their workload. To evaluate the unavailability dates of the lecturers and the critical lecturers in *Group Allocation Stage*, the critical lecturer is not allowed to take more off weeks in the semester period. The list of critical lecturers will then be processed in stage I *Lecturer Grouping Stage* to be assigned into different lecturer groups.

4.3 Two-Stage Heuristic Algorithm

In this study, a two-stage heuristic algorithm is deployed to solve the multi-location lecture timetabling problem. The stages in the heuristic algorithm are listed as follows and a diagram for the process is illustrated in Figure 4.1:

- (i) Lecturer Grouping Stage, and
- (ii) Group Allocation Stage

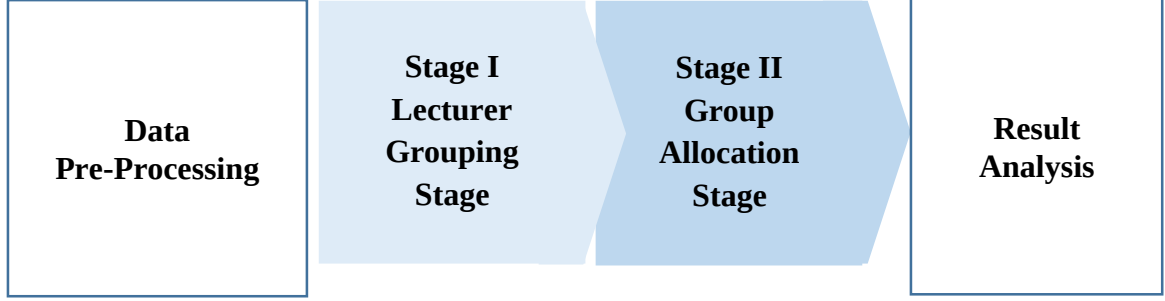


Figure 4.1: Stages in the Two-Stage Heuristic Algorithm.

Stage I: The *Lecturer Grouping Stage* clusters all the lecturers with respect to cohorts into the different groups with a simple rule that each group contain a unique set of the teaching lecturers. This is to prevent the timetable from having a conflict during the *Group Allocation Stage*.

Stage II: The *Group Allocation Stage* allocates each lecturer group into a set of academic weeks that is generated at prior stage. At this point, each lecturer in a group is determined by their unavailability dates and then assigned to the particular weeks. This stage ensures that all the allocated weeks for each group would not have any clashing problem when conducting the lecture sessions in multi-location.

4.4.1 Stage I: Lecturer Grouping Stage

In this stage, a set of lecturer groups is defined as $G = \{G_1, G_2, \dots, G_{|m|}\}$ and $|G| = g$ where $g < m$ to represent the number of lecturer groups. The *Lecturer Grouping Stage* first evaluates the ranking (criticalness) of the active lecturers. Each lecturer, l is then weighted and ranked before clustered into the lecturer groups, G . The “criticalness” of a

lecturer is determined prior in data pre-processing and used in the lecturer grouping algorithm. The more courses to be conducted by the lecturer, the higher the “criticalness” of the lecturer. The “criticalness” of the lecturer leads to the ranking of the student cohort, where the highest rank on student cohort and the most critical lecturer is assigned into lecturer groups first before others.

In the lecturer grouping algorithm, the ranking of a cohort is determined by the total of criticalness of the teaching lecturers, l in each cohort respectively. Later, the most critical cohort will first be considered in the assignment of the *Lecturer Grouping Stage*. In addition, the higher criticalness of the lecturers in that cohort is then first allocated into the lecturer groups, G .

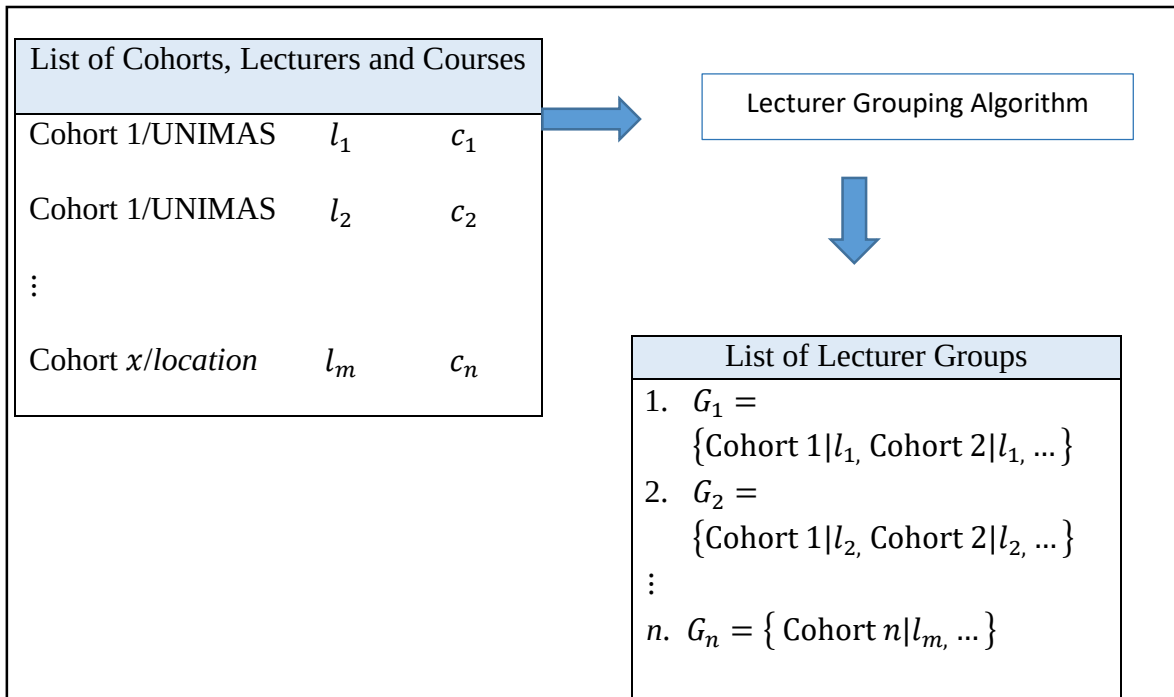


Figure 4.2: Lecturer Grouping Stage input output flow.

These lecturers are clustered to the same group if the conditions as defined in the following are met; Figure 4.2 shows the process of Stage II lecturer grouping algorithm:

- (1) There is no common lecturer in each group.
- (2) Teaching slot type of the course.

From the equation (4.2), a course needs a 14 hours lecture session for a week when conducting the lecture sessions in an off-UNIMAS campuses. Since each student cohort can be assigned with different set of courses to attend, in order to distribute the lecture sessions evenly without clash, a four lecturer groups are designed in this algorithm. Thus, across the 14 weeks of an academic semester, a total of four lecturer groups is used to assign the lecturers and no identical lecturer team should be presented in the same group. Each group represents the courses to be conducted at multi-location in a week. Besides that, for the lecture sessions that are conducted on either Saturday or Sunday, each group is allowed to be assigned two courses for one cohort but on the condition of no common lecturer in each group. Hence, each course that is conducted in one day can be assigned to two of the groups. To illustrate the said situation, Figure 4.3 shows the example of group assignment based on the teaching slot type of the courses.

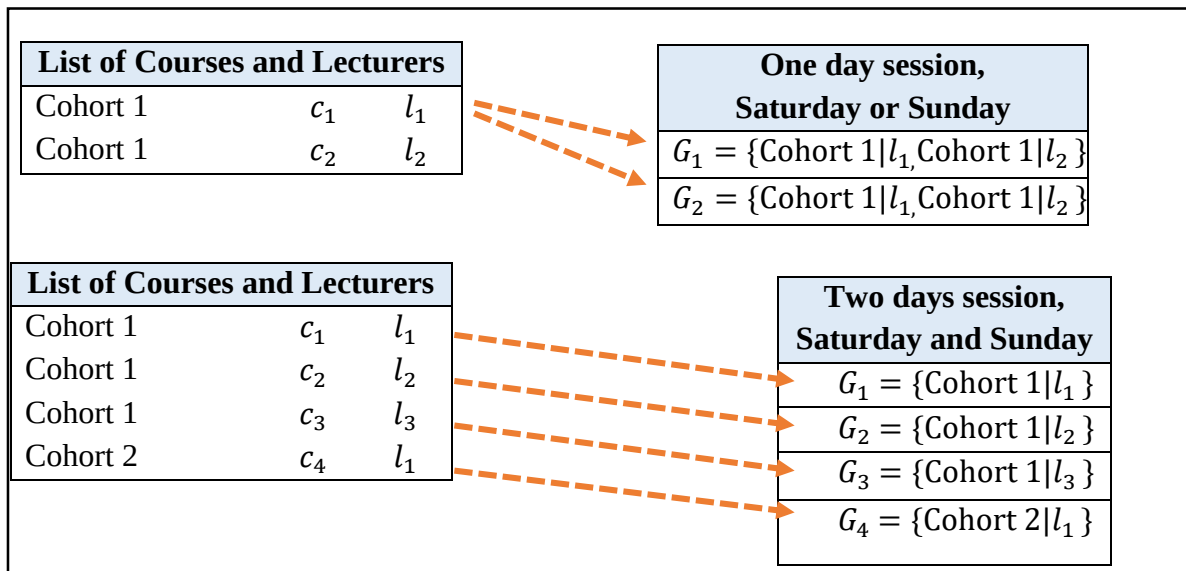


Figure 4.3: Group assignment based on teaching slot types.

The operational constraints such as a location or a course at one time and teaching slot types on location based are satisfied using the lecturers grouping stage. This result is then processed by *Group Allocation Stage*. Algorithm 4.1 explains the *Lecturer Grouping Stage* in the first stage of the proposed two-stage heuristic algorithm. Figure 4.4 illustrates the flow of lecturer grouping stage.

Algorithm 4.1: Lecturer Grouping Stage

```

Find all critical lecturer,  $l$  ranking
Compute each cohort,  $c$  ranking
Arrange  $c$  ranking in descending order
FOR  $i = c$  to 1
    Arrange  $l$  ranking in descending order
    FOR group,  $g = 1$  to  $n$ 
        Check group with  $l$ , set  $found$ 
        IF  $found = false$  then
            UPDATE the  $g$  list with  $l$ 
        ELSE
            CONTINUE to next  $g$ 
        ENDIF
    ENDFOR
ENDFOR
ENDFOR

```

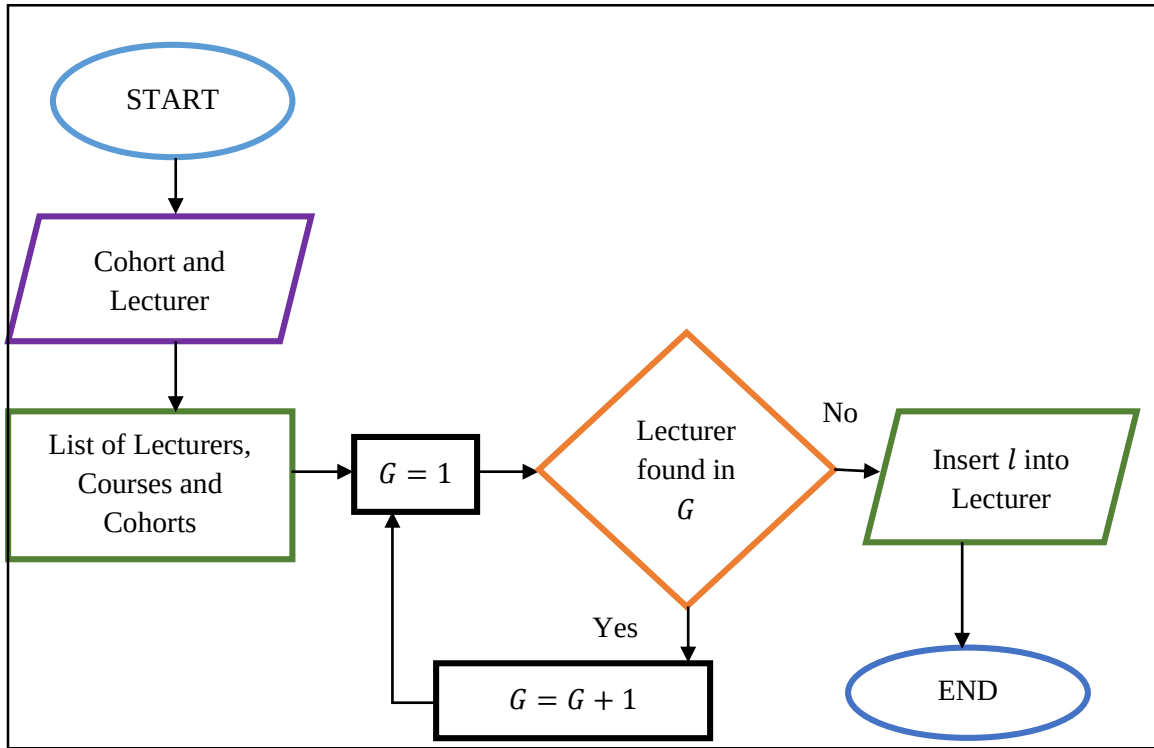


Figure 4.4: Flowchart of the lecturer grouping stage.

The Big O notation can be calculated from Stage I. The sequence of statements in the first loop executes N times, which N denotes the problem size of lecture timetabling. In the next sequence of statements, the If-Then-Else statements are nested looped within four groups. Assuming both If-Then-Else statement is $O(1)$, the total time for Stage 1 is $N * O(1)$, which is $O(N)$ overall.

4.4.2 Stage II: Group Allocation Stage

The *Group Allocation Stage* is ready to allocate the group of lecturers into a set of academic weeks throughout the semester. In this stage, the timeslot for each course can be allocated and the team-teaching slot for the lecturers can be assigned.

This stage starts from weighting each of the groups with unavailability dates of the lecturers in a set of academic weeks respectively. Next, a matching matrix is used to identify the availableness of the groups with n number of academic weeks, *week* in this *Group Allocation Stage*. This stage generates a $week \times g$ matrix with elements of 0 and 1. The element 0 represents that group is available and able to conduct a lecture session without the conflict of unavailability dates of the lecturers. On the other hand, the element 1 represents that the group is unavailable on that particular week and should not be assigned into that week. Figure 4.5 shows the example on the matrix performed when comparing the unavailability dates of the lecturers with the list of lecturer groups.

Each group is then weighted according to the total number of the unavailability dates of the lecturers and rearranged in weight size in a descending order. Within a set of academic weeks throughout the semester, the groups are assigned into the weeks by round robin optimisation. From this step, the *Group Allocation Stage* is ready to generate the timeslot allocation with the matrix from the unavailability dates of the lecturers.

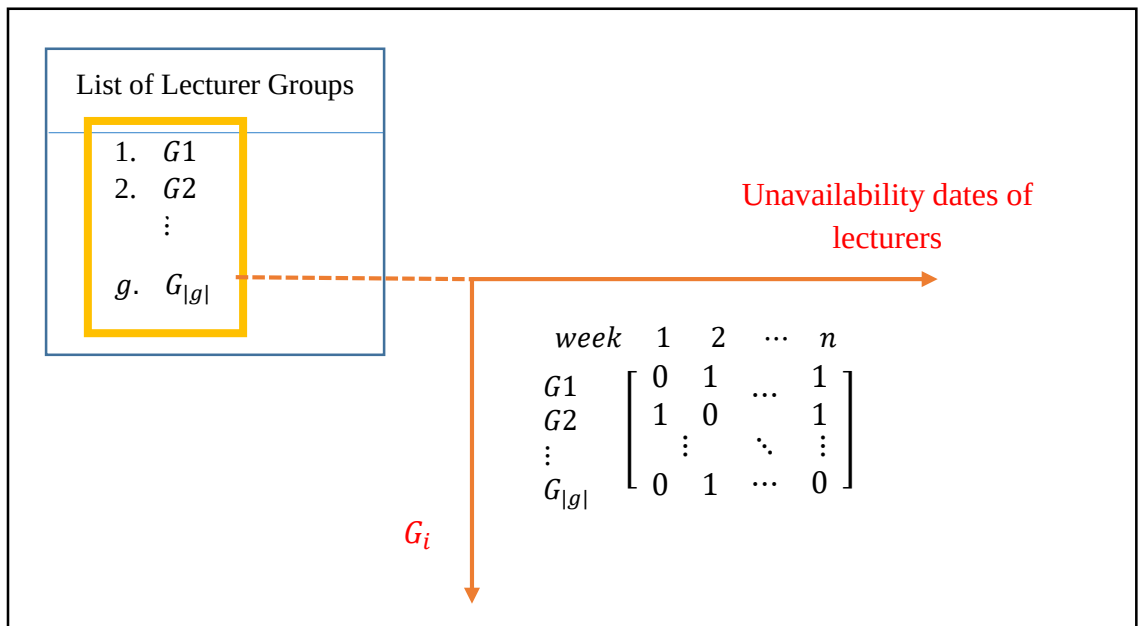


Figure 4.5: Generating a matrix in Group Allocation Stage.

It starts with the group with most unavailability dates of the lecturers. Then, the algorithm goes through the round robin optimisation with the matrix to allocate the groups. Figure 4.6 illustrates the example process of the group allocation stage.

The criteria to assign the timeslot to a lecturer group are listed as follow:

- (1) The group allocation is in round robin optimisation.
- (2) The available timeslot can be assigned to any one of the lecturer groups.
- (3) The unavailable timeslot should not be assigned to the lecturer groups which have unavailability dates on it.
- (4) The available timeslot can only be assigned to one of the lecturer groups.

Due to the round robin optimisation, the lecturer group with less timeslot allocation need to be considered first before assigning another lecturer group. The week 13 and week 14 are break into two days (Saturday and Sunday) is due to one more lecture sessions for a day need to conduct for UNIMAS slot type. Algorithm 4.2 explains the steps in *Group Allocation Stage* on a set of academic weeks for a semester.

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
G_1	Available	Unavailable	Available	Available	Occupied	Available	Available	Available	Occupied	Available	Available	Available	Occupied	Available
G_2	Unavailable	Occupied	Available	Available	Available	Occupied	Available	Available	Available	Occupied	Available	Available	Occupied	Available
G_3	Available	Available	Occupied	Available	Available	Available	Occupied	Available	Available	Available	Occupied	Available	Available	Occupied
G_4	Available	Unavailable	Available	Occupied	Available	Available	Available	Occupied	Available	Available	Available	Occupied	Available	Occupied
Available timeslot Occupied timeslot Unavailable timeslot														

Figure 4.6: Process of the Group Allocation Stage with a list of lecturer groups.

Algorithm 4.2: Group Allocation Stage

lct = list of lecturers in a group, g

FOR $week = 1$ to 14

 FOR $g = 1$ to n

 Compare *unavailability date* of lct with $week$

 IF $found = true$ then

 UPDATE *unavailable date* to g respectively to $week$

 ENDFOR

 ENDFOR

```

Compute total unavailable date in each g
Arrange g total unavailable date in descending order
FOR week = 1 to 14
    SET g = week % n
    Compare unavailable date to week, set found
    IF found = false then
        UPDATE week with the g
    ELSE
        CONTINUE to next g
ENDFOR

```

Figure 4.7 and Figure 4.8 show the processes of checking of unavailability and timeslot assignments for all the lecturer groups. After the *Group Allocation Stage*, the stage proceeds with team-teaching slot assignment process for all the courses and the lecturers individually. From this process, the one day or two consecutive days of lecture sessions are assigned based on their teaching slot types. Hence, a feasible timetable is generated at the end of the two-stage heuristic algorithm. Figure 4.9 shows the processes of timeslot assignment to each course and each lecturer in the multi-location.

The purpose of Big O notation from Stage II is to check the unavailability dates of lecturers for four different lecturer groups. Therefore, the first loop executes M times for checking each lecturer with their unavailability dates, where M denotes as total number of lecturer, the complexity of first loop would be $O(M)$. The next loop is allocating the four groups in round robin optimisation within 14 weeks. Thus, only one group is allocated in a week, the statement would be $O(1)$. The total time for Stage II is $O(M)$ for the worst-case statements.

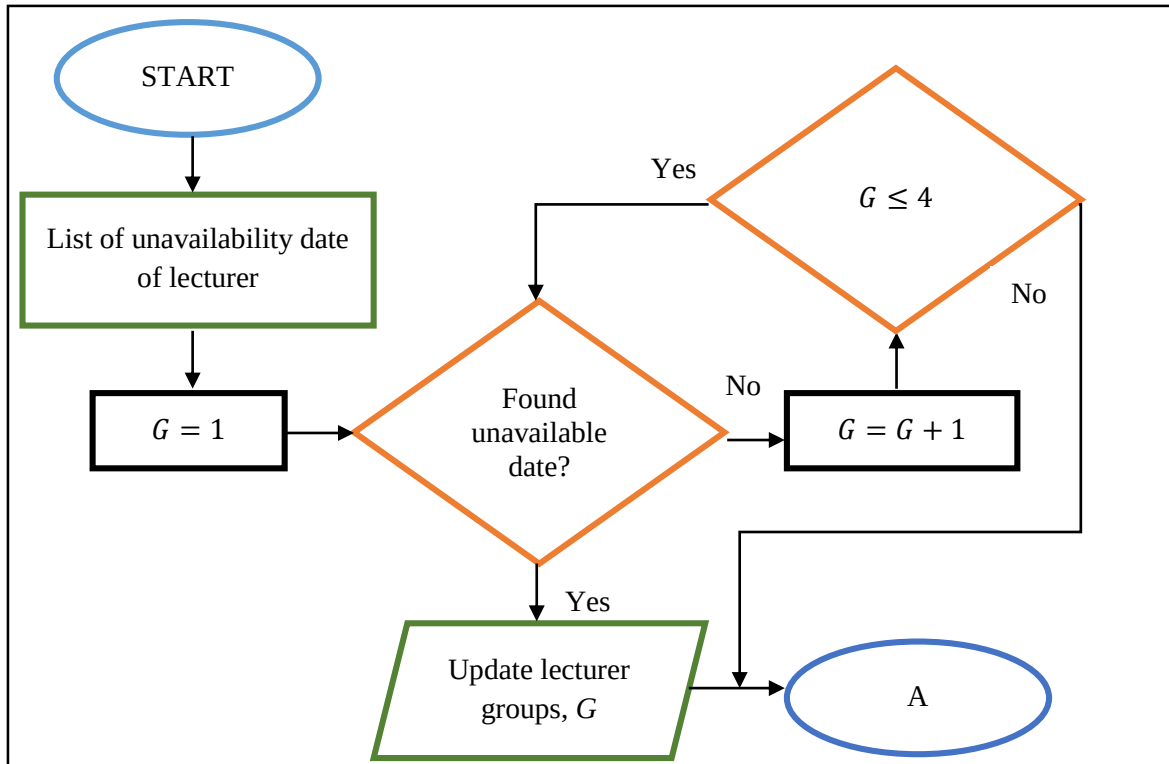


Figure 4.7: Flow chart of lecturer unavailability checking.

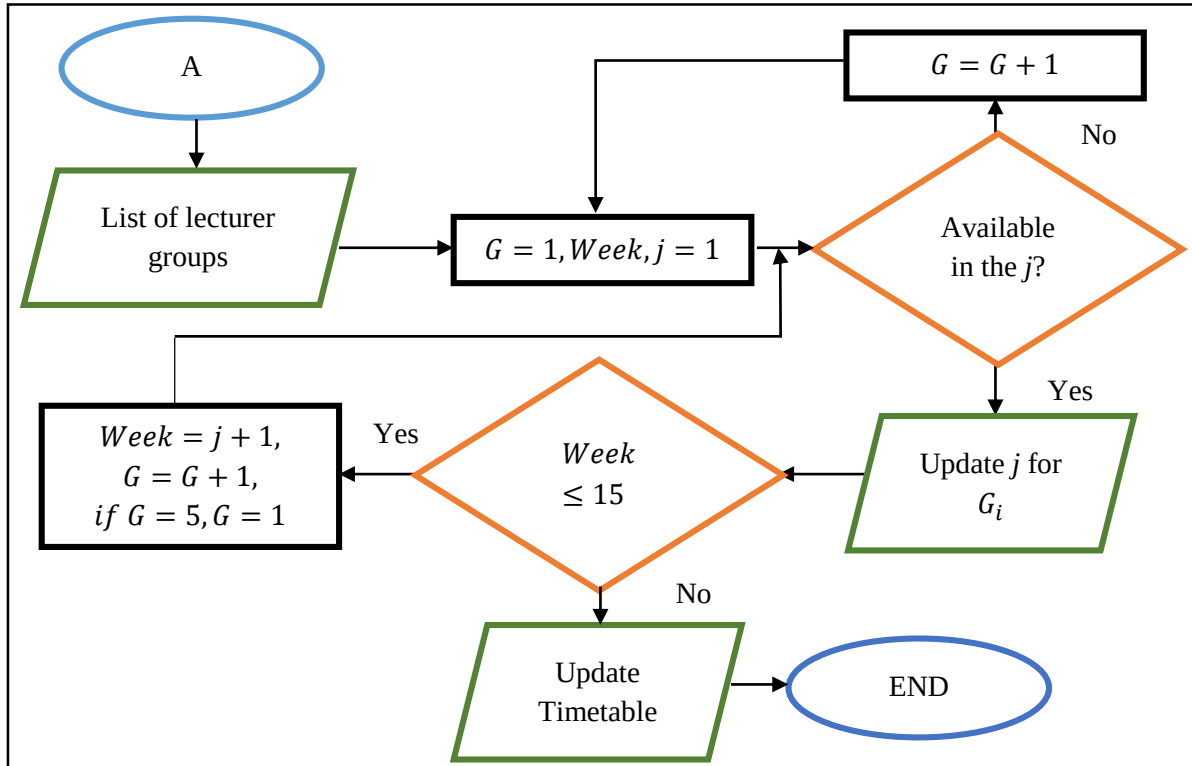


Figure 4.8: Flowchart of group allocation stage.

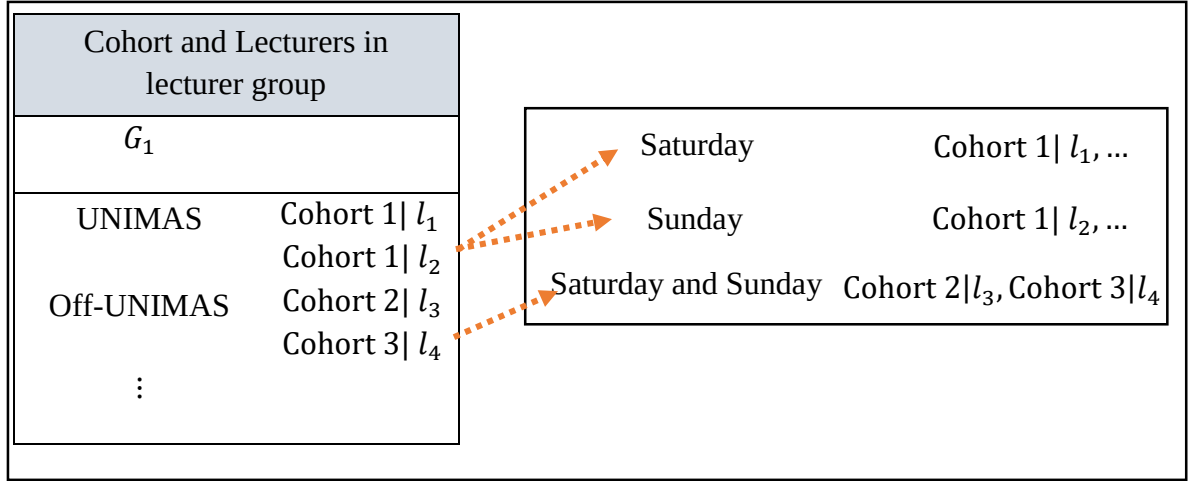


Figure 4.9: Timeslot assignment for lecture sessions for each course.

4.5 Summary

The heuristic algorithm is designed to solve the multi-location timetabling problem that can be achieved in two stages: *Lecturer Grouping Stage* and *Group Allocation Stage*, where each stage can fulfil different set of constraints and optimise the allocation method. The *Lecturer Grouping Stage* identifies the criticalness of the cohorts and the lecturers before being clustered into different lecturer groups with no common lecturers. Thus, this assures that each lecturer can conduct the lecture session in multi-location without a timetabling clashing problem. The *Group Allocation Stage* assigns and allocate the timeslot for the generated lecture groups in round robin optimisation. At the end of the two-stage heuristic algorithm, the lecture sessions that are to be conducted at multi-location are assigned to each course and each lecturer.

CHAPTER 5

RESULT AND ANALYSIS

5.1 Introduction

The proposed two-stage heuristic algorithm is validated by using the real data sets provided by MSc HRD in FCSHD. The timetable schedule for the lecturers who have been involved in conducting the coursework courses is analysed and explained in this chapter. Next, an analysis on the quality of the timetable is carried out by examining the distribution of lecture sessions and workloads of lecturers in UNIMAS and off-UNIMAS campuses. Furthermore, this chapter also includes the sensitivity analysis on the various constraints on this multi-location timetabling problem.

5.2 Dataset Description

Two real semester datasets (2017/2018-1 and 2017/2018-2) were collected from FCSHD to validate the proposed solution. The real timetable solution from these datasets are analysed and compared with the timetable result from the simulator. The size of problem for dataset 2017/2018-1 is relatively smaller as compared to dataset 2017/2018-2. This is because the MSc HRD program has initially been offered at off-UNIMAS campuses such as Kuala Lumpur (KL) and Sibul starting from the academic year 2017/2018. Therefore, a new cohort is introduced in each location. Subsequently, the number of courses and lecturers increased in semester 2017/2018-2.

Due to the travelling budget limitation, each lecture session for courses at off-UNIMAS campuses have increased to two consecutive days in a week or 14 hours per week. Table 5.1 shows the general information of each dataset.

Table 5.1: Dataset general information.

Datasets	2017/2018-1			2017/2018-2		
Location	KL	Sibu	UNIMAS	KL	Sibu	UNIMAS
Number of Lecturers	17			20		
Number of Cohorts	1	1	4	2	2	4
Number of Courses	3	3	8	4	3	11
Number of Locations	3			3		
Lecture Hours Per Sessions	6 (UNIMAS), 14 (KL, Sibu)					

5.3 Computational Analysis

In this section, computational analysis on the timetable solutions generated by the simulator is carried out. The analysis includes the comparison between the timetable solutions produced by the two-stage heuristic algorithm and the current timetabling practice. The simulator is coded with PHP web-based programming language and MYSQL database.

5.3.1 Overall Solution

Table 5.2: Comparison of method.

Dataset	2017/2018-1		2017/2018-2	
Method	Current practice	Two-stage	Current practice	Two-stage
Consider unavailability dates	×	√	×	√
Balance distribution of teaching schedule	×	√	√	√
Balance workload	√	√	√	√
Feasible solution	√	√	√	√
Course-based timeslot	√	√	√	√
Lecturer-based timeslot	×	√	×	√

The comparison of both datasets is reviewed by current practice and the two-stage heuristic algorithm is analysed as is shown in Table 5.2. The manually developed real timetable often need to be amended in order to avoid timetable conflict on unavailability dates of lecturers. From the analysis of the real timetable solution, for example dataset 2017/2018-1, it is found that there is an uneven distribution of teaching schedule among the courses that are offered to off-UNIMAS campuses. On the other hand, the real solution from dataset 2017/2018-2 presents a much even distribution of workload timetable to the lecturers than last semester. Both datasets are able to generate feasible timetables. However, frequent amendment on timetable solution is made due to the unavailability of the lecturers.

Meanwhile, the timetable solution that is generated from the simulator takes into consideration the unavailability dates of lecturers, and provide a better quality timetable solution in terms of distribution of teaching schedule and workload. The proposed algorithm is also able to provide a feasible lecturer-based timetable by assigning the timeslot for the respective lecturer. As compared to the current practice which is on course-based timetabling, the team-teaching lecturers have to discuss and plan their lecture sessions from the given timetable.

5.3.2 Computational Time

Table 5.3: Computational time comparison on both datasets.

Dataset	2017/2018-1		2017/2018-2	
Method	Current practice	Two-stage	Current practice	Two-stage
Stage 1	One to two weeks	0.41 seconds	One to two weeks	0.60 seconds
Stage 2		0.68 seconds		1.02 seconds
Total time		1.09 seconds		1.62 seconds

As discussed in previous chapter, the Big-O notation used by the heuristic algorithm is $O(N) + O(M)$, where N denotes the problem size of the dataset and M denotes the total number of lecturers. Table 5.3 shows the computational time taken by real solution and the two-stage heuristic algorithm on both datasets. The computational time that is taken by the proposed two-stage heuristic algorithm is bounded to be less than a minute in generating a timetable solution. Depending on the problem size for the number of courses and locations, the computational time varies for the timetabling problems from

different semester. Generally, the *Lecturer Grouping Stage* takes a longer time than the timeslot allocation stage. It is slightly time-consuming when the *Lecturer Grouping Stage* loops through several possible groups and uses priority rules to allocate the busiest lecturers in groups first.

5.3.3 Two-Stage Heuristic Algorithm

To illustrate the data processing on the algorithm of Stage I *Lecturer Grouping Stage* and Stage II *Group Allocation Stage*, Figure 5.1 and Figure 5.2 show the results from the processes of the proposed two-stage heuristic algorithm.

As in Figure 5.1, the process in Stage I ensures that each lecturer has not been allocated more than once in each group, which each lecturer group contains no common lecturers. However, the courses at UNIMAS campus are conducted separately in two different groups because there is only a day of lecture session in a week (UNIMAS slot type), for example the course with student cohort s_{11} is scheduled at two groups, G1 and G2. From the discussion in Chapter 3, the lecturer set L can be denoted as $L = \{l_A, l_B, l_C, \dots, l_m\}$ where m represents the total lecturers. Meanwhile the cohorts of students S can be denoted as $S = \{s_{11}, s_{12}, s_{13}, \dots, s_{pq}\}$ where p represents the location and q as cohort intake of students.

This information then undergoes a matrix performed in Stage II to determine the availability of lecturers in the weeks as illustrated in Figure 5.2. The available group is allocated first in round robin optimisation with trial and error method to ensure an evenly distribution of lecture sessions in the semester. An evenly distribution of lecture sessions can increase the students' performance in their academic. The unavailability of a lecturer is considered in Stage II in order to generate a clash-free timetable.

Stage I Lecturer Grouping Stage				
	Group One (G1)	Group Two (G2)	Group Three (G3)	Group Four (G4)
2017-2018-1	l_I, s_{23}	l_B, s_{23}	l_F, s_{23}	l_I, s_{13}
	l_F, s_{13}	l_D, s_{13}	l_E, s_{13}	
	l_E, s_{31}	l_I, s_{31}	l_L, s_{31}	
	l_B, s_{14}	l_C, s_{14}	l_A, s_{14}	
	l_J, s_{11}	l_N, s_{14}	l_O, s_{14}	
	l_H, s_{12}	l_K, s_{13}		
	l_M, s_{14}	l_G, s_{12}		
	l_P, s_{13}	l_Q, s_{11}		
2017/2018-2	l_R, s_{12}	l_H, s_{12}	l_H, s_{12}	
	l_H, s_{21}	l_C, s_{21}	l_F, s_{21}	
	l_G, s_{13}	l_D, s_{13}	l_E, s_{13}	
	l_C, s_{14}	l_A, s_{14}	l_B, s_{14}	
	l_P, s_{31}	l_J, s_{31}	l_I, s_{31}	
	l_K, s_{11}	l_L, s_{11}	l_M, s_{13}	
	l_Q, s_{13}	l_G, s_{22}	l_T, s_{14}	
	l_N, s_{14}	l_I, s_{13}	l_K, s_{12}	
	l_M, s_{11}	l_O, s_{14}		
	l_S, s_{12}	l_E, s_{12}		
		l_P, s_{11}		

	UNIMAS slot type
s_{1i}	UNIMAS
s_{2i}	Kuala Lumpur
s_{3i}	Sibu

Figure 5.1: Process of Stage I Lecturer Grouping Stage.

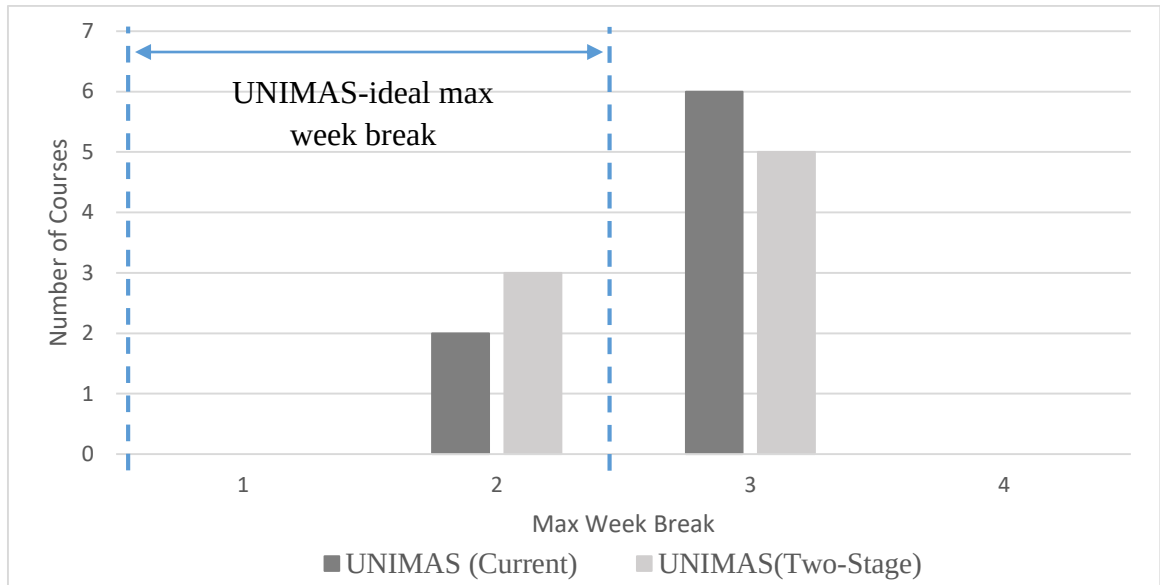
Stage II Group Allocation Stage																
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14		
G1																
G2																
G3																
G4																
Timeslot	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4	G1	G2	G3	G4

	Allocation for UNIMAS group		Allocations for group (UNIMAS and off-UNIMAS)
	Unavailable lecturers		

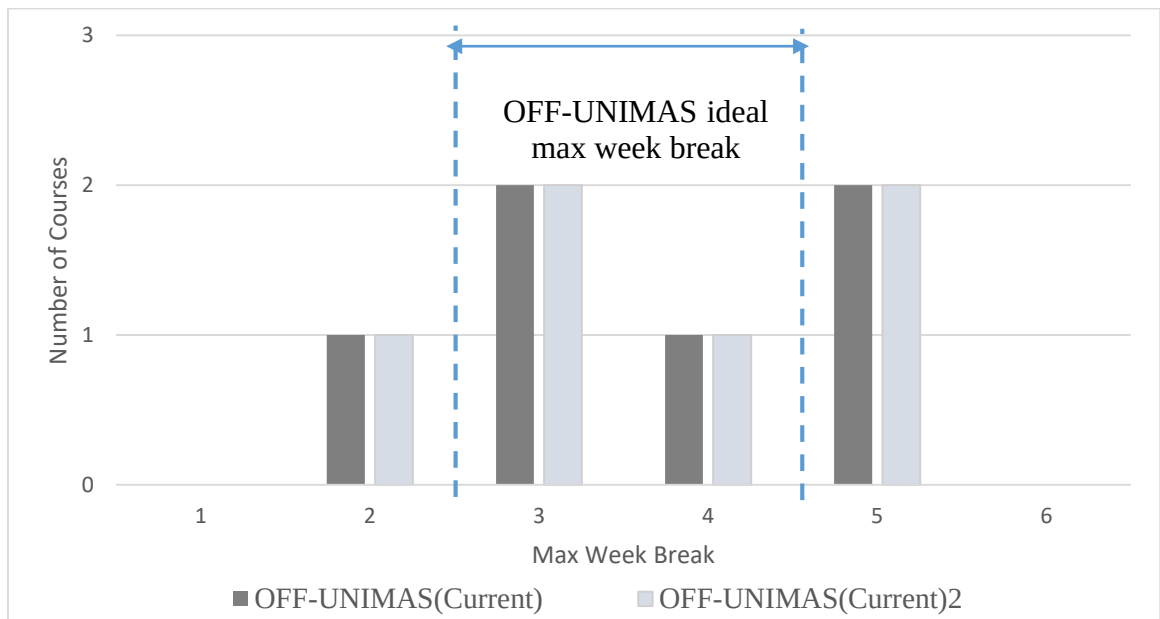
Figure 5.2: Process of Stage II Group Allocation Stage.

5.3.4 Distribution of Teaching Schedule

All constraints that are defined in this research are fulfilled at two stages of heuristic algorithm and a no clash solution is generated from the simulator. The ideal distribution for lecture sessions in UNIMAS is one to two weeks; meanwhile in off-UNIMAS campuses the breaks are three to four weeks. This is to ensure an evenly distributed timetable that is based on the courses and increase academic performance by the students. Figure 5.3 and Figure 5.4 show the comparison of a maximum week break between the real solution and the two-stage heuristic for dataset 2017/2018-1 and 2017/2018-2.

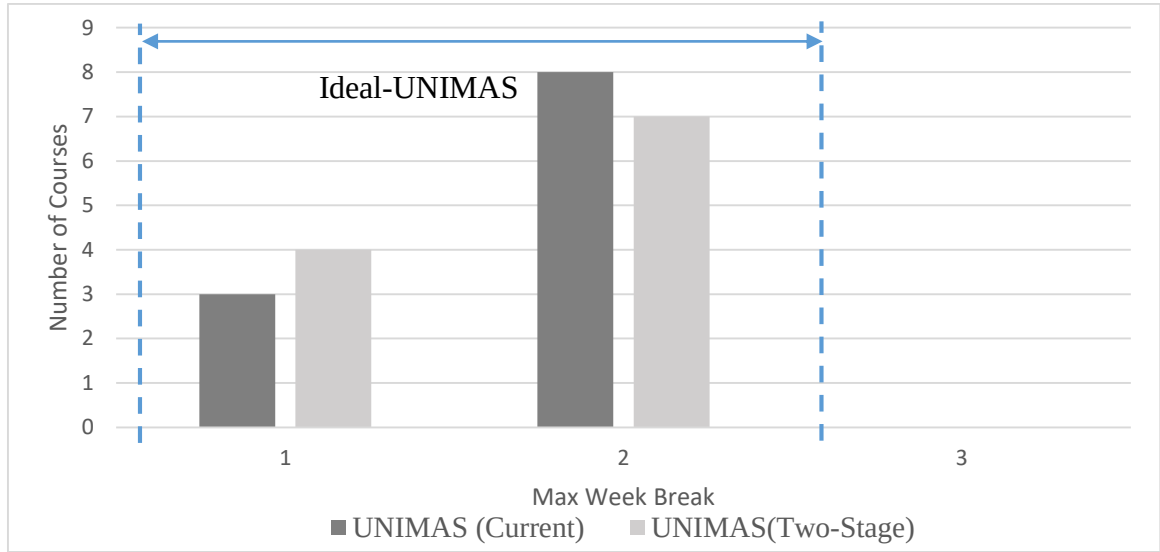


(a) UNIMAS current VS UNIMAS two-stage

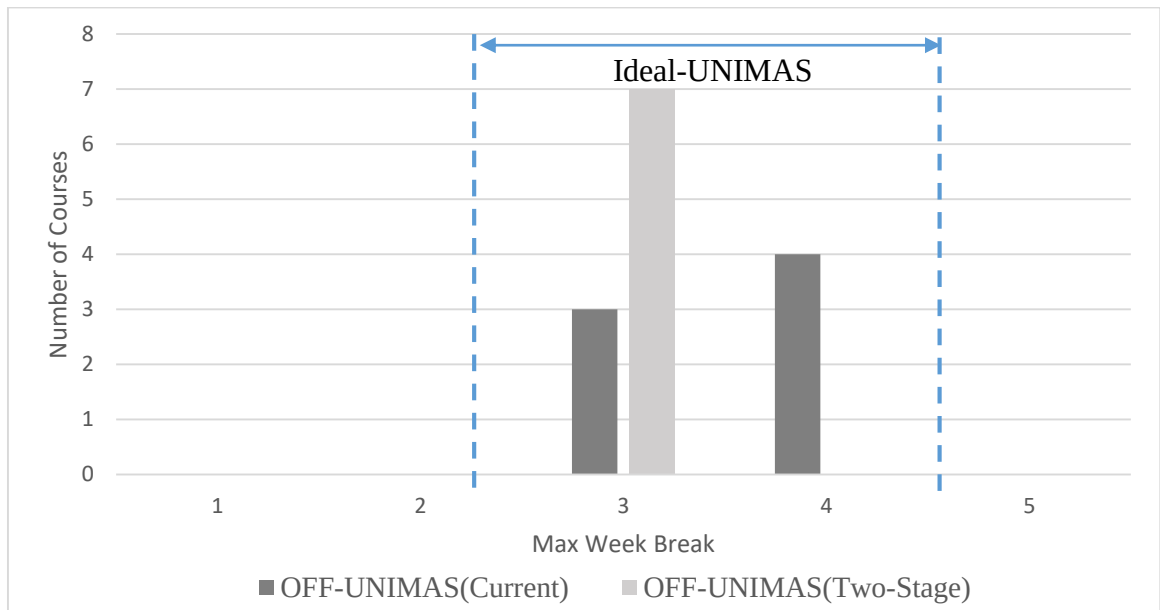


(b) OFF-UNIMAS current VS OFF-UNIMAS two-stage

Figure 5.3: Teaching schedule comparison for dataset 2017/2018-1.



(a) UNIMAS current VS UNIMAS two-stage



(b) OFF-UNIMAS current VS OFF-UNIMAS two-stage

Figure 5.4: Teaching schedule break comparison for dataset 2017/2018-2.

The two-stage heuristic algorithm allocate a better ratio of ideal distribution, which is 64.2% and 100% on lecture sessions in dataset 2017/2018-1 and dataset 2017/2018-2 respectively. By comparing the different percentage in both solutions, it can be observed that the two-stage heuristic algorithm offers a better distribution allocation on lecture sessions. The heuristic manages to produce an increment of five out of 14 courses or 28.5% that are within the range of ideal distribution as opposed to current practice in the dataset 2017/2018-1 which the current practice gives 35.7% ideal distribution ratio as shown in Figure 5.3. Some courses in UNIMAS have a maximum of three week break which is due to an addition of a mid-semester week between the lecture sessions.

As in Figure 5.4, the comparison of ideal ratio between the current timetabling practice and the two-stage heuristic algorithm shows that the method of distribution is very close to real solution. However, the solution from the two-stage heuristic algorithm achieves a higher number of courses in smaller “ideal max week” in dataset 2017/2018-2. The smaller maximum week break will results in more even schedule and higher academic performance of students on the courses. There is increasing of one and four courses in smaller maximum week break on UNIMAS and off-UNIMAS slot type respectively by heuristic algorithm as compared to current timetabling practice. The timetable solution generated by the simulator shows that the smaller the maximum week break, the better the quality of the timetable. From the result for dataset 2017/2018-2, the ideal smaller week break gap that achieved by simulator solution has increased to 36.4% or four out of 11 courses for UNIMAS campus and 100% or seven out of seven courses for off-UNIMAS campuses. This shows that the two-stage heuristic algorithm is capable to generate an equally or better timetables as compared to the current practice.

5.3.5 Distribution of Workload on Team-Teaching

Team-teaching, mostly pair-teaching has been practiced for MSc HRD program. This is to cope with limited lecturer manpower to conduct the courses at different locations. In team-teaching basis, the lecturers share equivalent amount of lecture sessions between the team. Due to the course-based real timetable solutions, the team-teaching lecturers have to discuss and plan their individual timetable after the timetable has produced. Therefore, there is no data on individual teaching's workload distribution from real timetable for comparison analysis. By using two-stage heuristic algorithm, the distribution of lecturer duty can be further analysed and validated on the amount of workload for each individual lecturer. The summary on the amount of workload is illustrated as Table 5.4.

Table 5.4: Summary of workload on team-teaching.

Course	Total number days of lectures			
	Team-teaching		No Team-Teaching	
	UNIMAS	Off-UNIMAS	UNIMAS	Off-UNIMAS
1	3 - 4	2 - 4	7	6
2	6 - 8	4 - 6	14	12
3	9 - 12	6 - 8	21	18

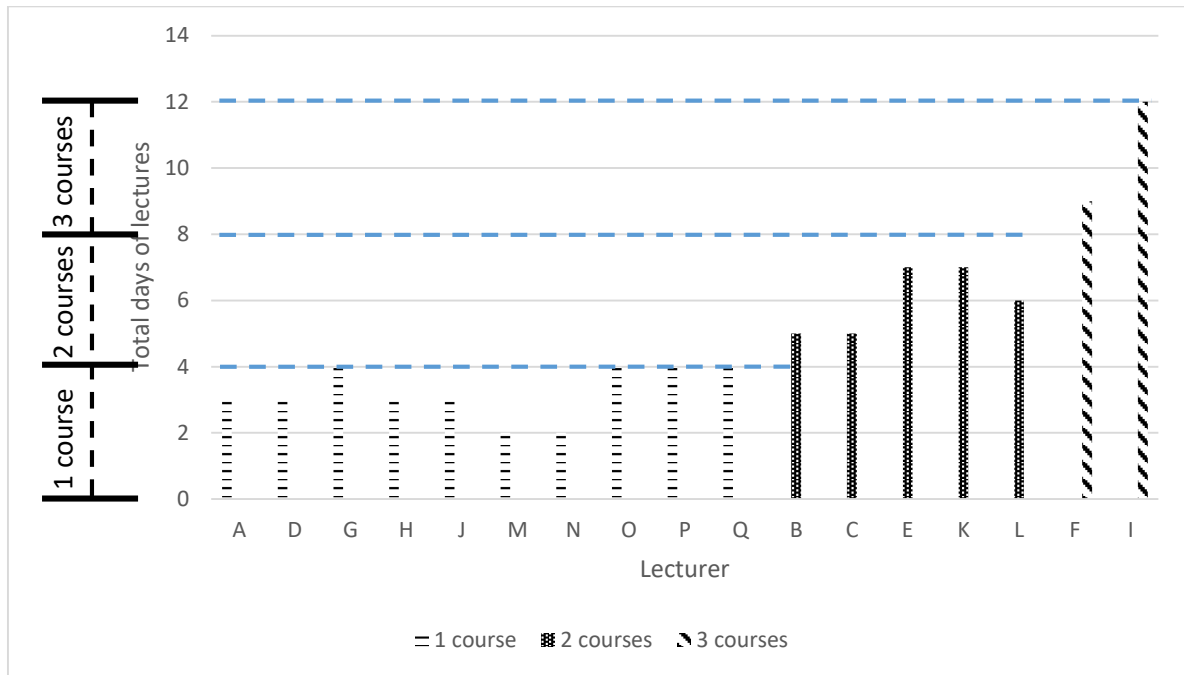


Figure 5.5: Distribution of workload on dataset 2017/2018-1.

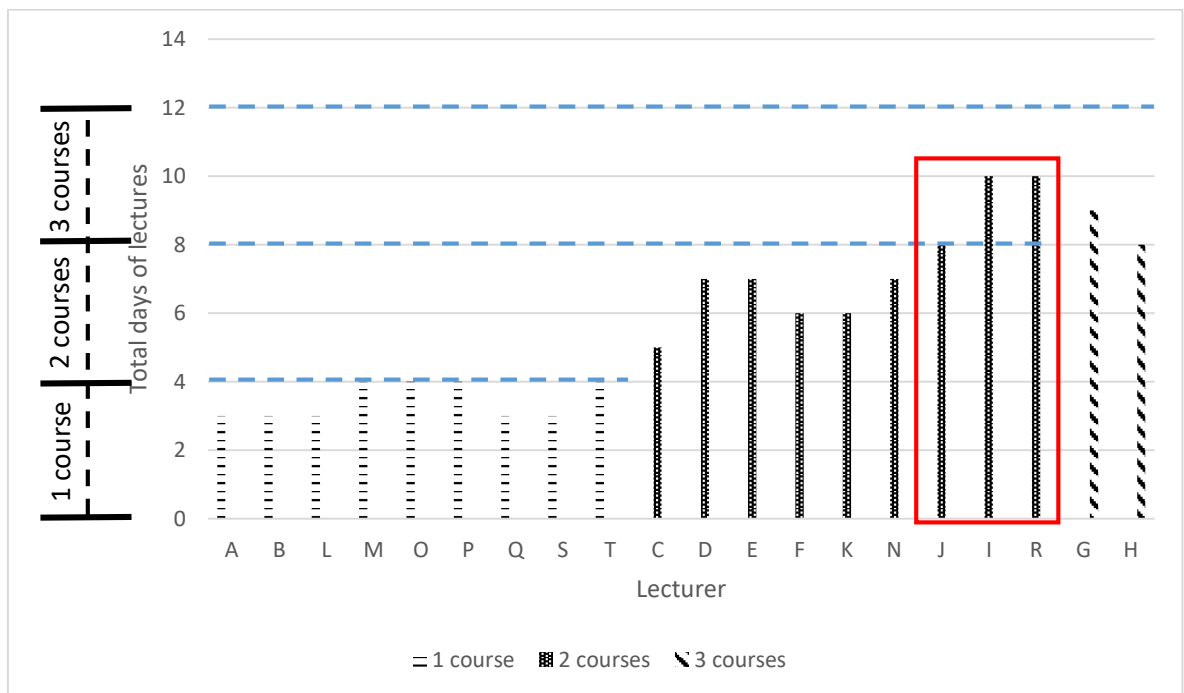


Figure 5.6: Distribution of workload on dataset 2017/2018-2.

Figure 5.5 and Figure 5.6 show the graphs on the distribution of workload that are analysed from the generated timetable solution from the simulator on the provided datasets. In Figure 5.5, all of the lecturers are team-teaching on a course and share an equivalent amount of lecture sessions on the courses. In dataset 2017/2018-2 as shown in Figure 5.6, there is a mixture of team-teaching and individual teaching (no team-teaching) basis.

From the analysis of Table 5.4, the lecturers who are team-teaching on one course at UNIMAS and off-UNIMAS campuses are conducting a total of two to four days of lecture sessions. For team-teaching on two courses, the lecturers are conducting a total of four to eight days of lecture sessions. Meanwhile, for team-teaching on three courses, the lecturers are conducting a total of six to twelve days of lecture sessions. Hence, these analysis show that the workload on team-teaching has been distributed fairly to all lecturers as is described in Table 5.4.

In Figure 5.6, the lecturer 'I' and 'R' are conducting a ten days of lecture sessions in a semester; meanwhile, the lecturer 'J' is conducting an eight days of lecture sessions, which is within the distribution range of workload for team-teaching on two courses. This is because lecturer 'I', 'J' and 'R' have a total of two courses to be conducted and one of them is off-UNIMAS campus based course that does not have team-teaching. Thus, lecturer 'I' and 'R' will have higher workloads than the rest of the lecturers as they have to conduct a six-days of lecture sessions for one course. Therefore, these lecturers will have higher workloads than the rest of the lecturers who are team-teaching on the courses at UNIMAS and off-UNIMAS campuses.

5.3.6 Unavailability dates of lecturer

In order to study the impact of unavailability dates of a lecturer on the simulator, the heuristic algorithm is tested with a set of unavailability dates on robustness. This is a real constraint when generating a timetable solution for MSc HRD program. Since this research does not have the real data on unavailability dates of lecturers which can only be obtained upon request from coursework coordinator, a set of dummy unavailability dates of lecturers are tested.

The timetable solution remains the same teaching team and different teaching slot type for UNIMAS and off-UNIMAS campuses. The amount of unavailability dates of a lecturer is increased until the simulator is unable to generate a feasible solution. In order to generate a feasible timetable solution, the simulator is not allowed to allocate the timeslots on the weeks when the lecturers are out of duty.

To illustrate the situation on shifting the conflict in a timetable solution, Table 5.5 shows the application of the allocated timeslots and the unavailability dates of a lecturer. In order to study the impact of unavailability of lecturers on the timetable solution, an ideal distribution of a teaching schedule of a lecturer from each category is analysed, e.g. lecturers from team-teaching on one course, two courses and three courses. In this case study, a consecutive weeks of unavailability dates is applied to each case study. The analysis on the distribution of teaching schedule takes lecturer 'A', 'B' and 'I' from dataset 2017/2018-1 (DS1) and lecturer 'A', 'C', 'R' with 'I' from dataset 2017/2018-2 (DS2) as subjects in this case study, where lecturer 'I' has one course that does not have team-teaching basis. Table 5.6 shows the summary of dataset on the lecturer unavailability and Figure 5.7 shows the impact of the unavailability dates on distribution of teaching schedule.

Table 5.5: Group allocation comparison with unavailability dates of lecturer.

Unavailability dates of Lecturers															
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Unavailable		G2			G1	G2									
Group allocation without unavailability dates of lecturers															
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Group 1															
Group 2															
Group 3															
Group 4															
Group allocation with unavailability dates of lecturers															
Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Group 1															
Group 2															
Group 3															
Group 4															

	Mid semester break		UNIMAS slot type
	Unavailable lecturers		Off-UNIMAS slot type

Table 5.6: Summary of lecturer unavailability and its category.

Category	Dataset 2017/2018-1	Dataset 2017/2018-2
1 course	Lecturer 'A'	Lecturer 'A'
2 courses	Lecturer 'B'	Lecturer 'C'
3 courses	Lecturer 'I'	Lecturer 'R'
2 courses (individual teaching on one course)	-	Lecturer 'I'

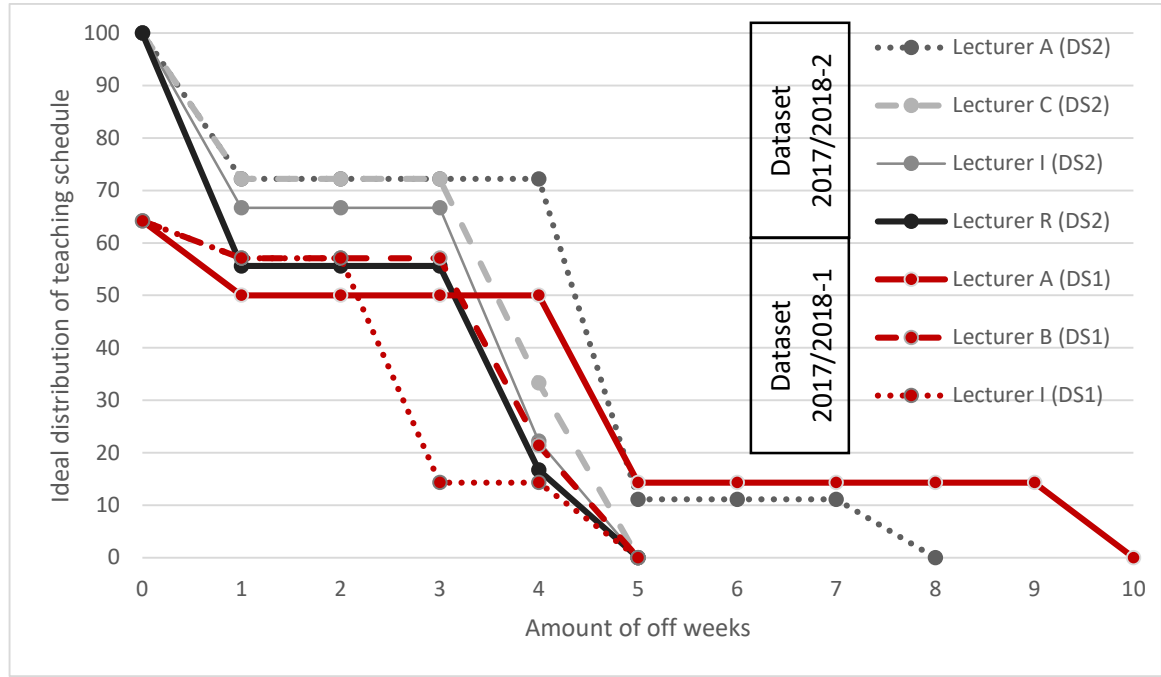


Figure 5.7: Ideal distribution of teaching schedule on unavailability dates.

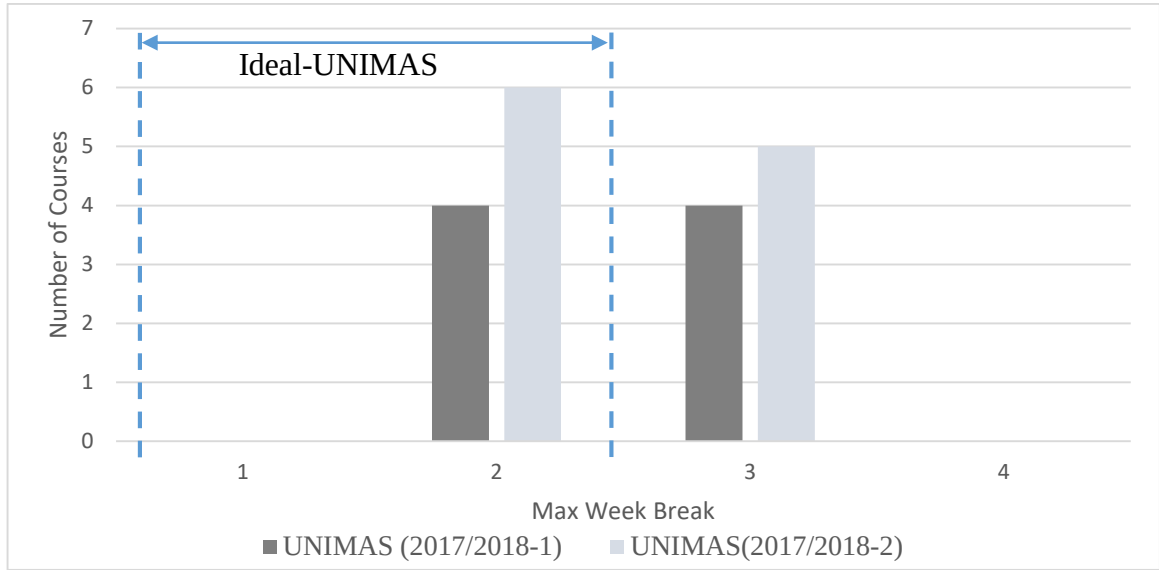
Based on Figure 5.7, the unavailability dates of a lecturer can result in decreasing the ideal distribution ratio in a teaching schedule. From the initial timetable solution that is analysed previously, the ideal ratios that have been achieved for dataset 2017/2018-1 and 2017/2018-2 are 64.2% and 100% respectively. The Figure 5.7 shows that, despite the increasing number of off-weeks of a lecturer in the simulator, the two-stage heuristic algorithm is still able to distribute the workload equally to all the lecturers, irrespective of the quality of the teaching schedule. The ideal ratio of the teaching schedule drops gradually after a consecutive of off-weeks of lecturers from each category is applied in the timetable solution. When five consecutive of unavailability are applied for the lecturers who are team-teaching on three courses, the timetable solution is still able to generate a feasible solution but with poor quality solutions. Thus, unavailability dates of a lecturer can have an effect on the overall quality of the timetable solution.

5.4 Sensitivity analysis

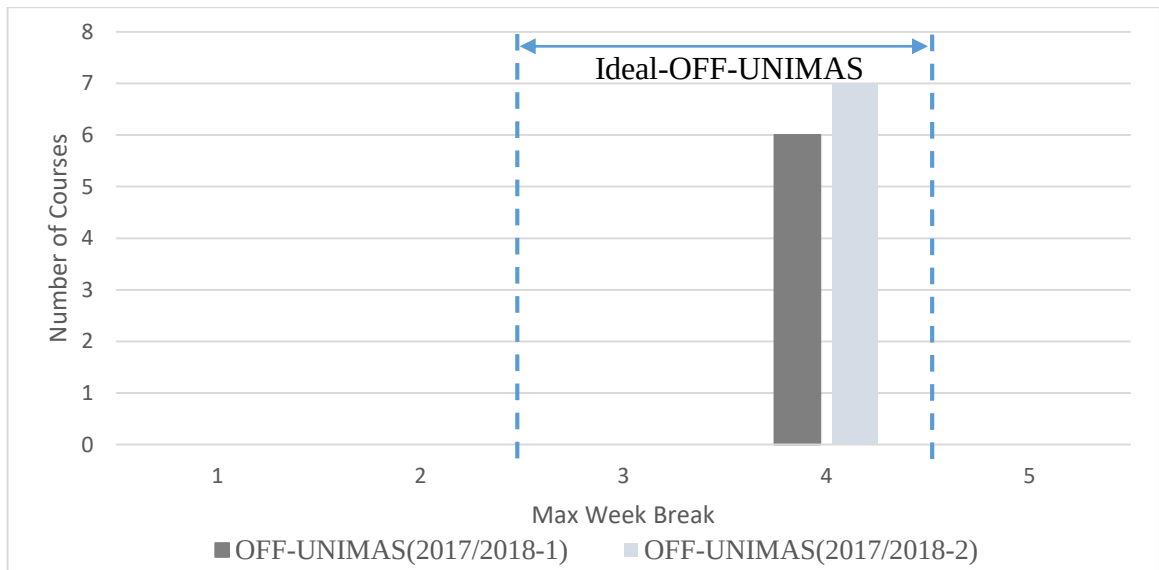
This section discusses the impact of change on the parameters of the proposed two-stage heuristic algorithm to observe the feasibility of this research. The same datasets of 2017/2018-1 and 2017/2018-2 are used in this sensitivity analysis on the same teaching slot type on UNIMAS slot type and off-UNIMAS slot type, and no team-teaching setting.

5.4.1 No Team-Teaching

In order to study the sensitivity analysis of no team-teaching, the constraint of team-teaching is being relaxed. Each course requires exactly a lecturer to conduct the lecture sessions throughout the semester. Thus, no secondary lecturer is allowed for the courses. Dataset 2017/2018-1 has been reduced to 10 lecturers for 14 courses while dataset 2017/2018-2 has been reduced to 15 lecturers for 18 courses. Consequently, a lecturer may be required to travel to multi-location throughout the semester to conduct the lecture sessions. To check the feasibility of the two-stage heuristic algorithm, the quality of the solutions is compared. Due to the decreasing number of lecturers for the courses, it is reasonable that the timetable solution may be unable to distribute the teaching schedule evenly for all the courses. Figure 5.8 shows the ideal ratio of maximum week break between lecture sessions for both datasets.



(a) No team teaching comparison for UNIMAS on dataset 2017/2018-1 VS dataset 2017/2018-2



(a) No team teaching comparison for off-UNIMAS on dataset 2017/2018-1 VS dataset 2017/2018-2

Figure 5.8: Teaching schedule comparison on no team-teaching.

Based on Figure 5.8, the dataset 2017/2018-1 and 2017/2018-2 have one or more lecturers who may conduct two or more courses in a semester. The quality of the solutions in terms of the distribution of the teaching schedule is affected. For dataset 2017/2018-1, the ideal ratio has an increment of +7.2% from the initial solution which is 64.2% from the simulator. However, for dataset 2017/2018-2, the ideal ratio has a decrease of -27.8% from the initial solution which is 100%. This is because lecturer ‘I’ conducts three courses at one time in dataset 2017/2018-2, whereas most lecturers only have two courses to conduct. These results show that the simulator is still able to generate a feasible solutions from the no team-teaching in the teaching schedule.

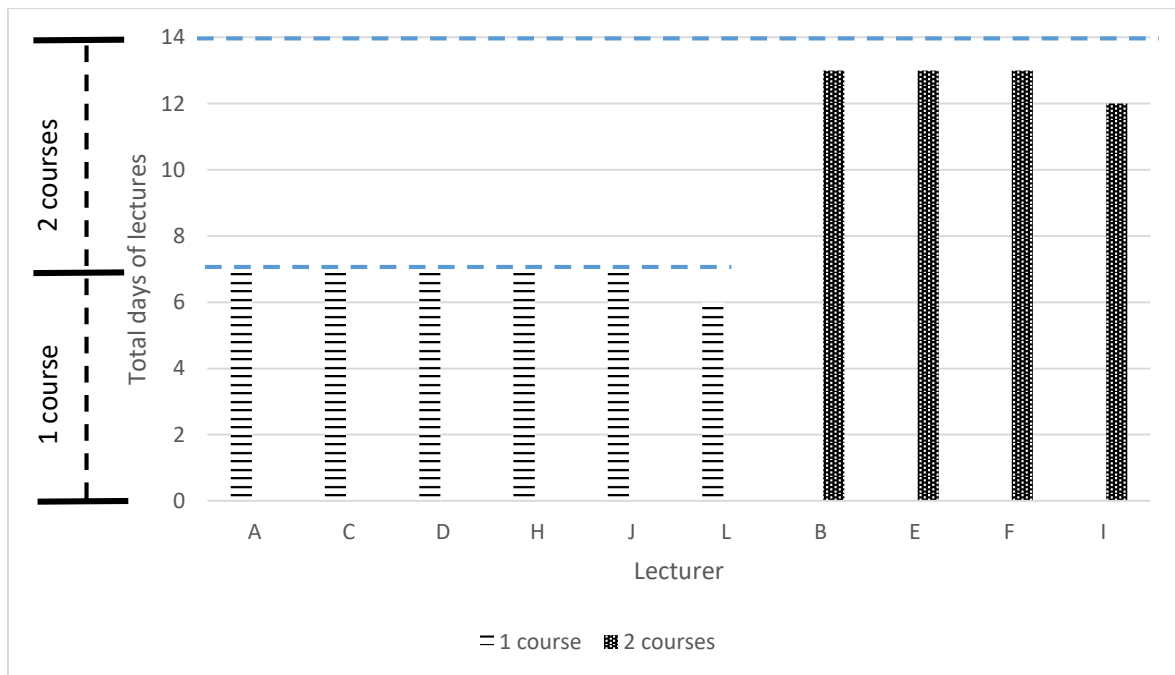


Figure 5.9: Workload on no team-teaching on dataset 2017/2018-1.

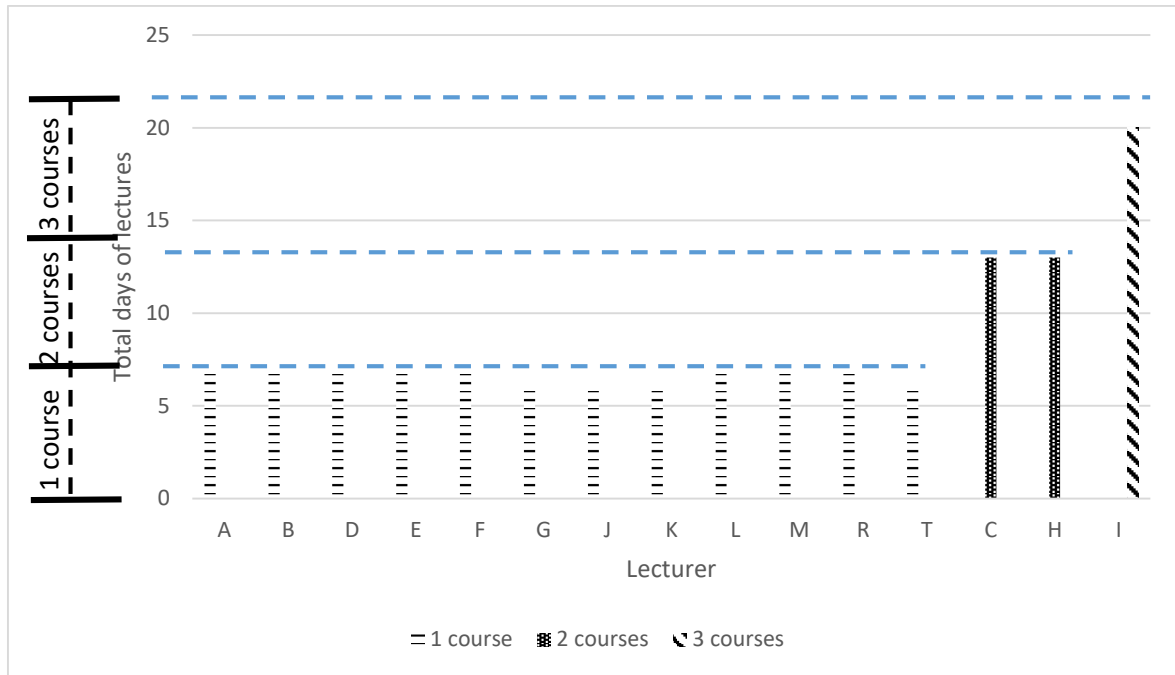


Figure 5.10: Workload on no team-teaching on dataset 2017/2018-2.

Table 5.7: Summary of workload on no team-teaching.

No team-teaching		
Course	Total number days of lectures	
	UNIMAS (one day of lecture session)	Off-UNIMAS (two days of lecture sessions)
1	7	6
2	14	12
3	21	18

From the analysis on Figure 5.9 and Figure 5.10, the distribution range of workload is within the range as summarised in Table 5.7. These analysis shows that the distribution of workload is valid for all the lecturers with no team-teaching. In this case study of no team-teaching on the courses, the algorithm is still able to generate a feasible and clash-

free timetable solution. Table 5.7 shows that to conduct one course at UNIMAS campus without team-teaching requires a total of seven days to fulfil the teaching hours; meanwhile, off-UNIMAS campuses require a total of six days to complete a course. In Figure 5.10, the lecturer 'I' can conduct three courses (two courses at UNIMAS campus and one course at off-UNIMAS campus) at one time at different teaching locations, thus the lecturer needs a total of 20 days to conduct in order to fulfil the total teaching hours for three courses. This result shows that no team-teaching on the courses is still able to generate a feasible timetable and the distribution of workload is within the desirable range.

5.4.2 Same Teaching Type Slot

In this case study of the same teaching type slot, all the lecture sessions that are conducted are within a week session or a consecutive two-day lectures session. In this subsection, a consecutive two-day lectures session per week for the courses is applied for UNIMAS campus and off-UNIMAS campuses as shown in Table 5.11; meanwhile, a one day lecture session per week for the courses is applied as shown in Figure 5.12. The timetable solution maintains the same teaching team and no unavailability dates of lecturers. Figure 5.11 and Figure 5.12 show the distribution of teaching schedules impact on the same teaching slot type for both data sets.

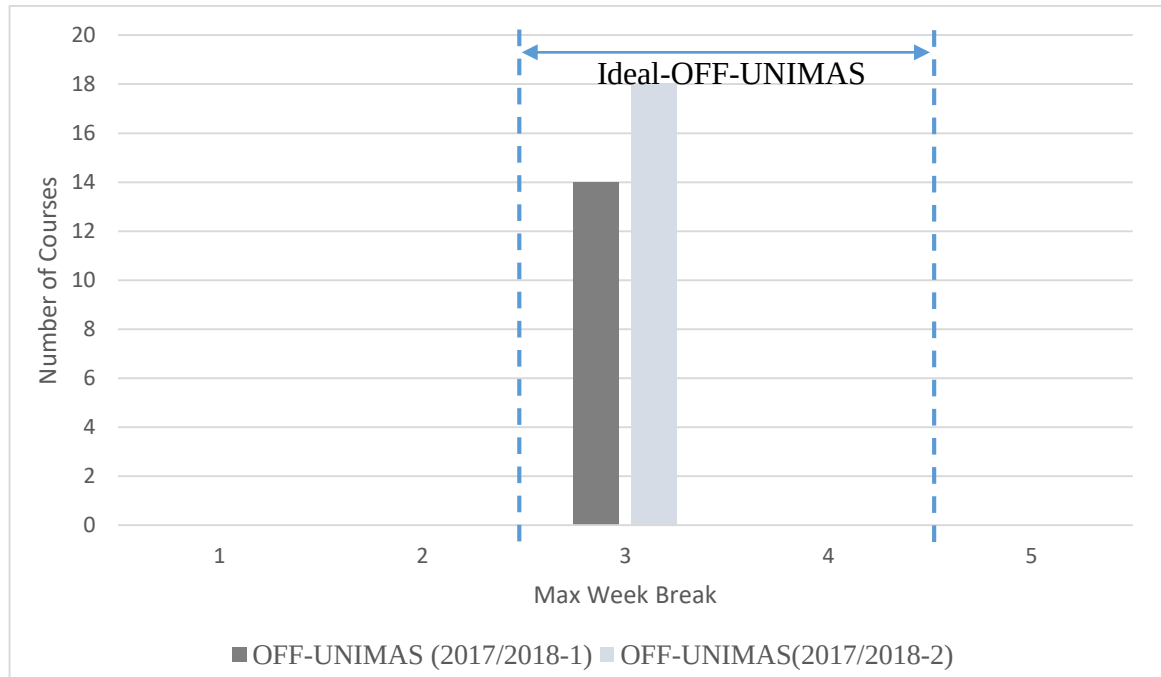


Figure 5.11: Teaching schedule comparison on off-UNIMAS slot type.

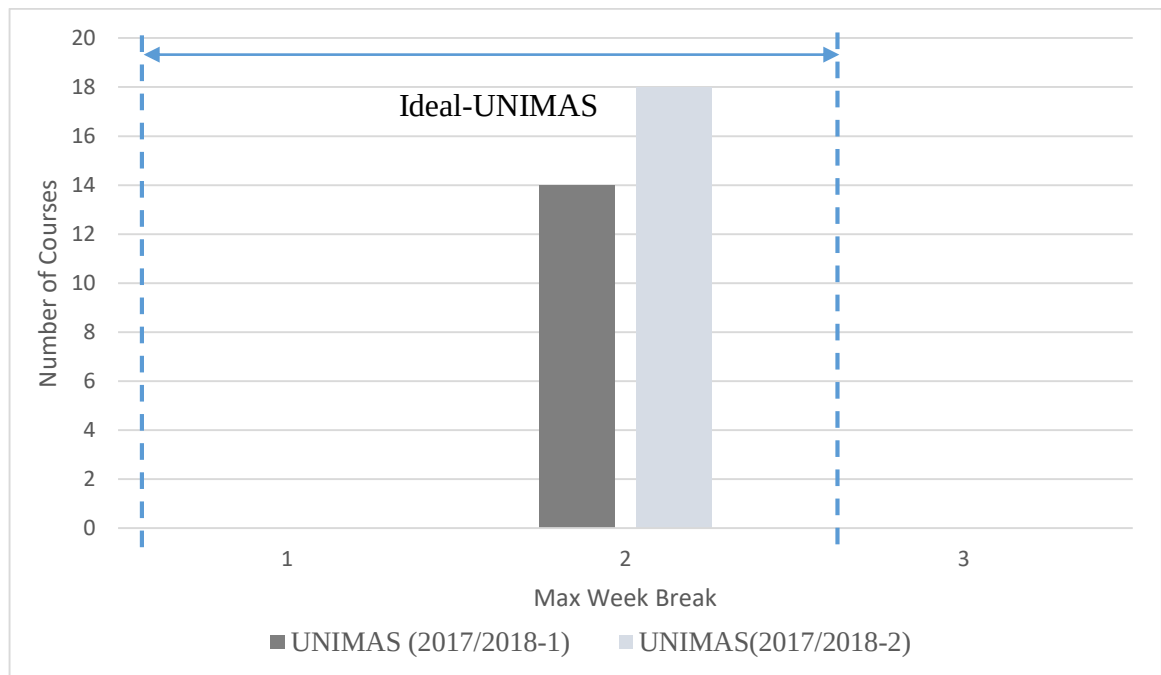


Figure 5.12: Teaching schedule break comparison on UNIMAS slot type.

Based on the Figure 5.11 and Figure 5.12, both solutions have achieved 100% ideal distribution of teaching schedule for all the courses based on UNIMAS slot type and off-

UNIMAS slot type. By applying the UNIMAS slot type and off-UNIMAS slot type respectively on dataset 2017/2018-1, there is an increase of 35.8% in the ideal ratio of teaching schedule for all teaching locations. Overall, the quality of the timetable solutions has improved when the same teaching slot type are applied to all the courses in UNIMAS and off-UNIMAS campuses.

In terms of workload distribution, the next analyses are conducted with a consecutive two-day teaching slot type and one day per week teaching slot type respectively. Based on the teaching slot types, Table 5.8 summarises the finding of total days of lectures to conduct for the courses.

Table 5.8: Summary of workload on same teaching slot types.

Same Teaching Slot Types				
Course	Total number days of lectures			
	A consecutive two-day lectures		One-day lecture per week	
	Team-teaching	No team-teaching	Team-teaching	No team-teaching
1	2 - 4	6	3 - 4	7
2	4 - 8	12	6 - 8	14
3	6 -12	18	9 - 12	21

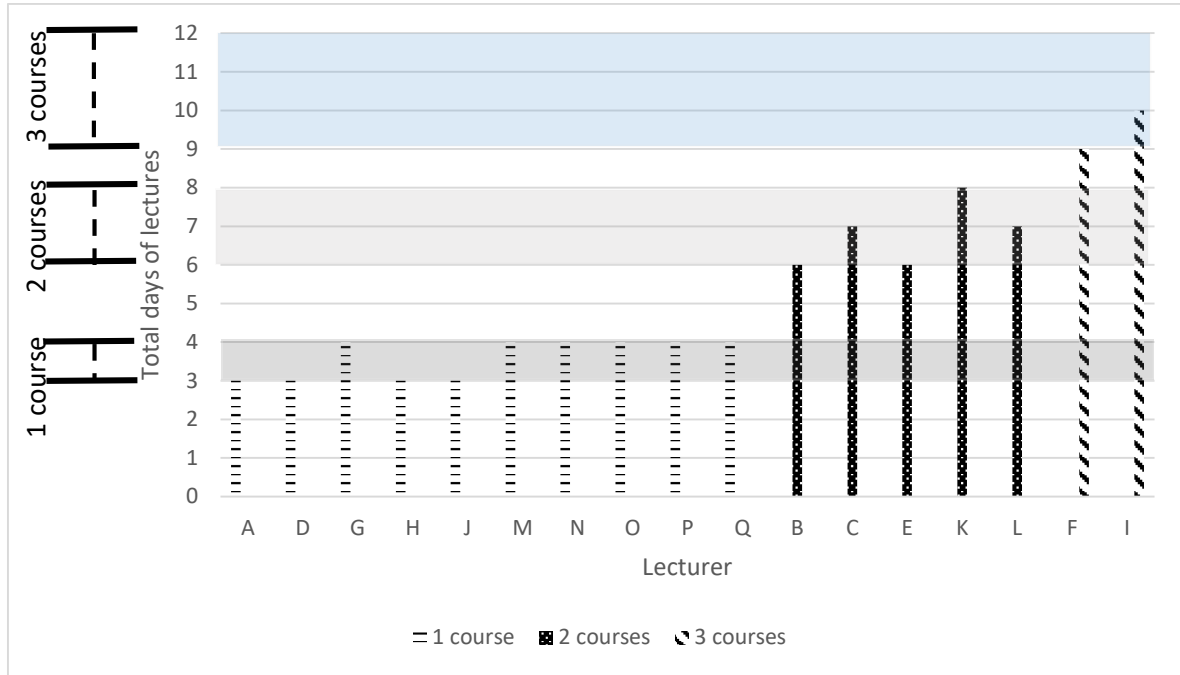


Figure 5.13: Distribution of workload on a one-day teaching slot type on dataset 2017/2018-1.

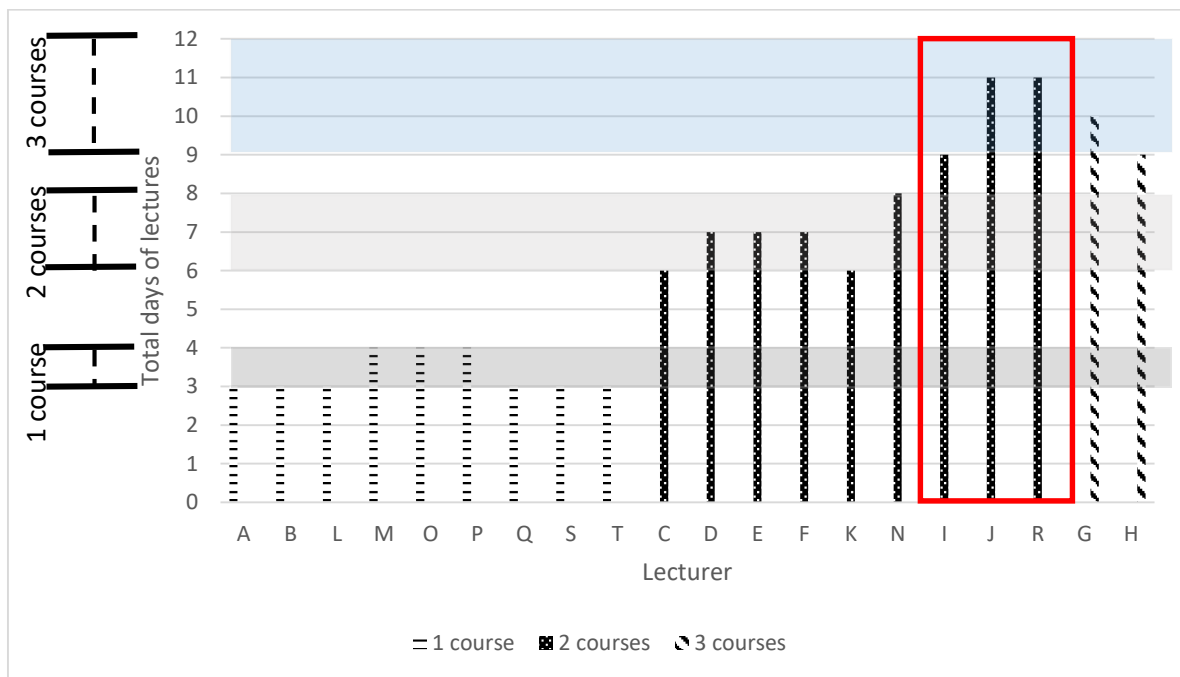


Figure 5.14: Distribution of workload on a one-day teaching slot type on dataset 2017/2018-2.

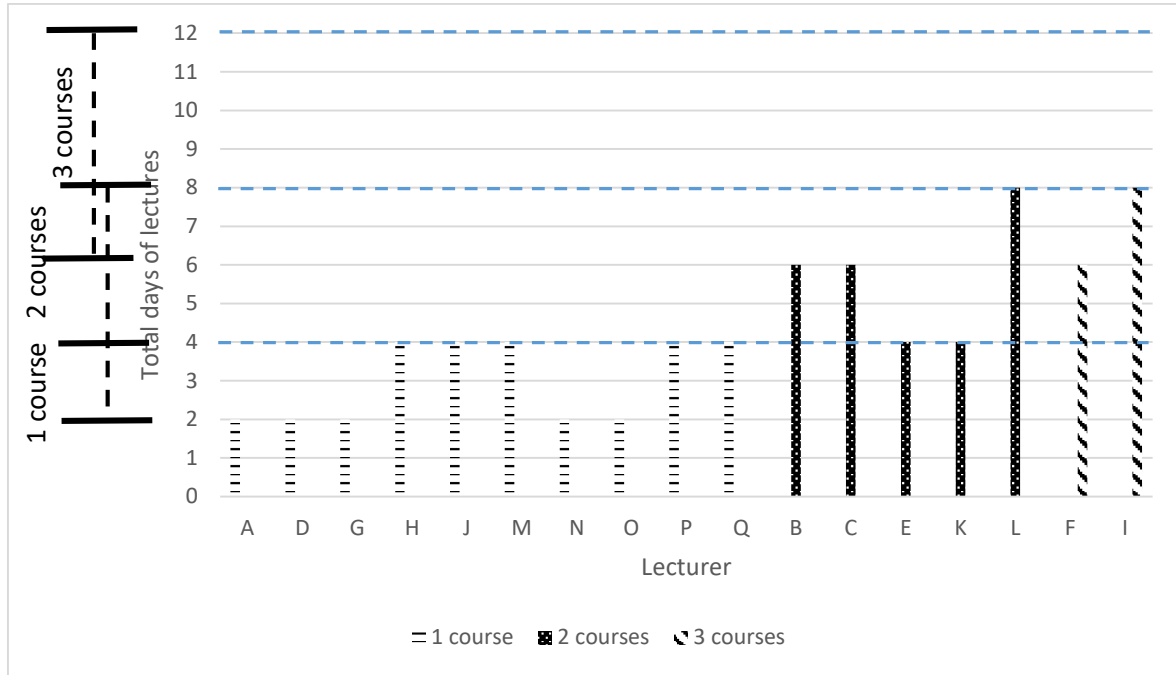


Figure 5.15: Distribution of workload on a consecutive two-day lectures on dataset 2017/2018-1.

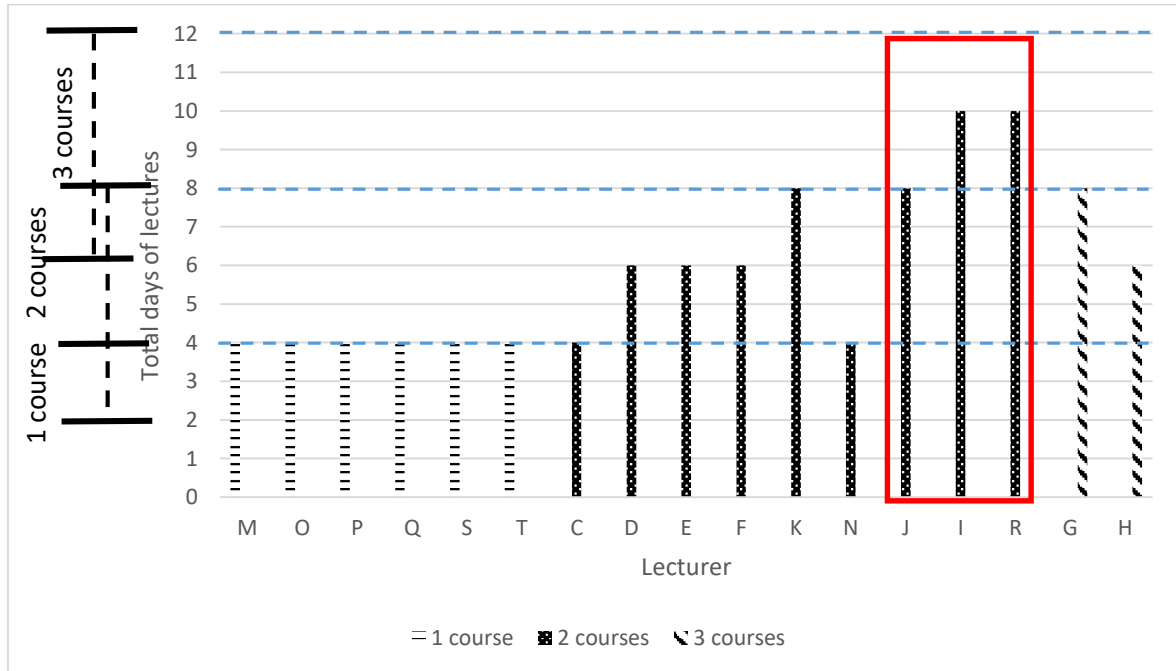


Figure 5.16: Distribution of workload on a consecutive two-day lectures on dataset 2017/2018-2.

From the analysis of a one-day teaching slot type (as shown in Figure 5.13 and Figure 5.14) and a consecutive two-day teaching slot type (as shown in Figure 5.15 and Figure 5.16), the results show all team-teaching lecturers share an equivalent amount of workload on the courses.

From the Figure 5.14 and Figure 5.16, each of the lecturers 'I', 'J' and 'R' has two courses where one is without team-teaching to conduct the lecture sessions. On the case study of one-day teaching slot type as shown in Figure 5.14, lecturer 'I' and 'R' have to conduct a total of 11 days (7+4) lecture sessions for two courses. This is because a total of a seven-days lecture sessions is needed to be conducted by a lecturer who is on a no team-teaching one-day teaching slot type; Meanwhile, for two-consecutive two-day teaching slot type as shown in Figure 5.16, lecturer 'I' and 'R' have to conduct a total of 10 days (6+4) lecture sessions due to a total of a six-days lecture sessions that are needed to be carried out on a consecutive two-days teaching slot type without team-teaching. Thus, lecturer 'I', 'J' and 'R' have the highest workload as compared to those lecturers who are team-teaching on two courses.

5.5 Summary

The two-stage heuristic timetable solutions that are generated by the simulator are compared with the real timetable solution that has been created using the current timetabling practices. The timetable solution that is generated by the simulator is a clash-free solution. Moreover, the simulator timetable solution has outperformed the current practice with fast computational time of less than one minutes. On the other hand, the real solution with the current timetabling practices can take one to two weeks. Subsequently, the real solution has a lot of clashing issues due to limited manpower resources and is

unable to cope with the unavailability dates of a lecturer. Frequent adjustments on timetable solution are required and thus requires a longer time to generate a feasible timetable solution.

In terms of the ideal ratio of teaching schedule, the simulator timetable solutions illustrate that the courses that fall within the acceptable range of maximum weeks break have been greatly increased. The simulator solution manages to allocate an ideal ratio of up to 64.2% for dataset 2017/2018-1. Besides that, the simulator solution can also fulfil the same value for dataset 2017/2018-1 up to 0% differ from the real solution.

In terms of the workload distribution, on team-teaching, the lecturers share equivalent amount of lecture sessions to conduct the courses. The timetable solutions from the simulator show that the workload is distributed equally and fall within the desired range of days of the lecture sessions for the lecturers.

A good quality of the solution is defined by a high ideal ratio distribution of lecture sessions and the distribution of workload for a lecturer is within the desired range. This research also focuses on the impact of unavailability dates of lecturer on the proposed two-stage heuristic algorithm. The results from the simulator show that a feasible timetable solution can be obtained despite the poor quality of the solution. The simulator is still able to generate a feasible solution even though a five-consecutive weeks of unavailableness of a lecturer are applied in this problem.

A series of sensitivity analysis are conducted on the proposed heuristic in terms of no team-teaching on courses and same teaching slot type. When applying the same teaching slot type to all the courses in all teaching locations, the ideal teaching schedule can achieve 100% on both datasets and the workload is distributed evenly for all lecturers.

When no team-teaching on the courses, as is limited by manpower resources, the ideal ratio of the teaching schedule increases up to 7.1% on dataset 2017/2018-1 while achieving 100% of ideal distribution of teaching schedule on dataset 2017/2018-2. To conclude, all these results demonstrate that the proposed solution is effective and robust in solving MSc HRD coursework timetabling problem in different scenarios.

CHAPTER 6

CONCLUSION

6.1 Introduction

This chapter concludes the overall implementation of this research project and it summarises the contributions of the present study to the related body of knowledge. Possible future work that can be done for this multi-location timetabling problem is also discussed.

6.2 Conclusion of Research

A case study of the multi-location coursework timetabling problem for the MSc HRD coursework programme in FCSHD has been investigated in this research. Current timetabling practices in MSc HRD coursework programme are solved by using the trial and error method. Thus, timetable adjustments and revisions are made through multiple times to solve the lecturer conflict issues. To cope with the increasing number of students and locations, a two-stage heuristic algorithm is designed in order to solve the multi-location coursework timetabling problem. To generate a timetable solution, this heuristic algorithm considers the constraints of multi-location and the lecture unavailability dates. Besides that, the increasing number of locations, students and courses to be conducted from time to time add to the timetabling difficulties in this problem.

The research objective of this study is to design and construct a computational bounded heuristic for multi-location timetabling problem. This research has analysed the requirement of this timetabling problem and then produce a heuristic algorithm to solve the problem. The automated timetabling assisted with the proposed heuristic algorithm can generate a clash-free timetable solution within a short time.

The main aspect of heuristic algorithm is due to its simplicity and ease to apply to generate a feasible solution within a short time. Prior to the heuristic algorithm, a data pre-processing phase is to verify and validate the data sets to generate a timetable solution without an error. The initial stage of the two-stage heuristic method is the *Lecturer Grouping Stage*, which clusters the lecturers into a set of lecturer groups. The following stage, the *Group Allocation Stage* assigns the lecturer groups into a set of timeslots of a 14 academic weeks. The *Group Allocation Stage* then assigns each course with its teaching slot type respectively in the timetable solution. The generated solution is a no-clash solution as each lecturer group in the *Lecturer Grouping Stage* would not have common team-teaching lecturers.

Real data sets from two semesters are collected from FCSHD and tested with the heuristic algorithm. The computational results show that the timetable solution that has been produced by the two-stage heuristic algorithm can perform similar or outperform the current timetabling practices. In term of computational time, at the most one minute is used to generate a feasible solution with no timetable adjustment needed. In term of conducting the lecture sessions at multi-location, each lecturer can only conduct at one location at one time. It can also provide a better performance on the distribution of lecture sessions for a course over a set of academic weeks. The ideal week break between the lecture sessions in UNIMAS main campus is one to two weeks, while the ideal maximum week break between the lecture sessions is three to four weeks for off-UNIMAS campuses. Besides that, the distribution of lecture duty has been evenly allotted in the timetable solution. In terms of the unavailability dates of a lecturer, the heuristic algorithm can cope with the availableness constraint to generate a feasible solution without any clashing issue. All

these results demonstrate that the proposed solution is effective and robust in solving multi-location timetabling problem for different scenarios.

The achievements of research objectives are accomplished in the end of this research and explained in Table 6.1.

Table 6.1: Objectives and achievements.

Objectives	Achievements
(i) To design a heuristic model for multi-location coursework timetabling problem.	A two-stage heuristic algorithm are developed to solve the multi-location coursework timetabling problem, namely <i>Lecturer Grouping Stage</i> and <i>Group Allocation Stage</i> .
(ii) To formulate a mathematical model of multi-location coursework timetabling problem.	Various hard constraints are defined on lecturers, students, total teaching hours for a course, different teaching slot types based on location and unavailability dates of lecturer. The soft constraints that defined in this problem are even distribution of timetable and lecturer duty in order to generate an optimal solution. The objective function of this problem is to minimise the lecture session gap of courses for the students.
(iii) To conduct sensitivity analysis on the proposed heuristic for multi-location coursework timetabling problem.	A series of sensitivity analysis is conducted on unavailability dates of lecturers, same teaching slot type and no team-teaching setting. The results demonstrate that the heuristic algorithm is effective and robust in solving multi-location problem in different scenarios.

6.3 Research Contributions

There are several contributions that this research has achieved upon its completion:

- (a) A fast response two-stage heuristic algorithm is developed to solve the multi-location lecture timetabling problem in FCSHD. This algorithm uses the trial and error method, and fast in getting a feasible solution.
- (b) *Lecturer Grouping Stage* and *Group Allocation Stage* are introduced in the two-stage heuristic algorithm. The first stage clusters all the lecturers with respect to cohorts into a set of teaching groups. Then, the second stage allocates the teaching groups from Stage I into a set of academic weeks throughout the semester.
- (c) The operational constraints in multi-location lecture timetabling problem such as unavailability dates of lecturers, different teaching slot type based on locations, team-teaching of lecturers, total teaching hours for a course to be fulfilled are considered in the heuristic algorithm.
- (d) Developed an automated system for the multi-location lecture timetabling method. An individual timetable with lecture sessions to be conducted for each lecturer is designed and planned.

6.4 Future Work

There are several aspects that could be investigated for the future development of this research. Firstly, the timeslot allocation in the *Group Allocation Stage* could be enhanced in order to improve the quality of the timetable solution by using allocation problem approach. Other than that, the designed automated timetabling is developed for lecture sessions only, automated timetabling for mid-term and final examination might be worthwhile to investigate as it completes all the schedules of a student's timetable. The

multi-location timetabling problem can be extended to more locations and conducted for a larger data set of students and courses.

REFERENCES

- Abdullah, S., Turabieh, H., McCollum, B., & McMullan, P. (2012). A hybrid metaheuristic approach to the university course timetabling problem. *Journal of Heuristics*, 18(1), 1-23.
- Adewumi, A. O., Sawyerr, B. A., & Montaz Ali, M. (2009). A heuristic solution to the university timetabling problem. *Engineering Computations*, 26(8), 972-984.
- Adriaen, M., De Causmaecker, P., & Berghe, G. V. (2003). Decentralised course timetabling in a large hierarchical organisation. *Proceedings MISTA*.
- Aladag, C. H., & Hocaoglu, G. (2007). A tabu search algorithm to solve a course timetabling problem. *Hacettepe Journal of Mathematics and Statistics*, 36(1), 53-64.
- Almeida, M. W. S., Medeiros, J. P. S., & Oliveira, P. R. (2015). Solving the academic timetable problem thinking on student needs. In *Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on* (pp. 673-676). IEEE.
- Al-Omari, H., & Sabri, K. E. (2006). New graph coloring algorithms. *American Journal of Mathematics and Statistics*, 2(4), 739-741.
- Burke, E. K., & Newall, J. P. (2004). Solving examination timetabling problems through adaption of heuristic orderings. *Annals of Operations Research*, 129(1-4), 107-134.
- Burke, E. K., & Petrovic, S. (2002). Recent research directions in automated timetabling. *European Journal of Operational Research*, 140(2), 266-280.

- Burke, E. K., Cowling, P., & Silva, J. L. (2001). Hybrid population-based metaheuristic approaches for the space allocation problem. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on* (Vol. 1, pp. 232-239). IEEE.
- Burke, E. K., Kendall, G., & Soubeiga, E. (2003). A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, 9(6), 451-470.
- Burke, E. K., Petrovic, S., & Qu, R. (2006). Case-based heuristic selection for timetabling problems. *Journal of Scheduling*, 9(2), 115-132.
- Burke, E., & Erben, W. (Eds.). (2003). *Practice and Theory of Automated Timetabling III: Third International Conference, PATAT 2000 Konstanz, Germany, August 16-18, 2000 Selected Papers* (Vol. 2079). Springer.
- Burke, E., & Rudova, H. (Eds.). (2007). *Practice and Theory of Automated Timetabling VI: 6th International Conference, PATAT 2006 Brno, Czech Republic, August 30-September 1, 2006 Revised Selected Papers* (Vol. 3867). Springer.
- Burke, E., Bykov, Y., & Petrovic, S. (2000). A multicriteria approach to examination timetabling. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 118-131). Springer, Berlin, Heidelberg.
- Chaudhuri, A., & De, K. (2010). Fuzzy genetic heuristic for university course timetable problem. *International Journal of Advances in Soft Computing and its Application*, 2(1), 100-121.
- Chowdhary, A., Kakde, P., Dhoke, S., Ingle, S., Rushiya, R., & Gawande, D. (2014). Timetable generation system. *International Journal of Computer Science and Mobile Computing*, 3(2), 410-414.

- 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.
- Dammak, A., Elloumi, A., Kamoun, H., & Ferland, J. A. (2008). Course timetabling at a Tunisian university: A case study. *Journal of Systems Science and Systems Engineering*, 17(3), 334-352.
- Dandashi, A., & Al-Mouhamed, M. (2010). Graph coloring for class scheduling. In *Computer Systems and Applications (AICCSA), 2010 IEEE/ACS International Conference on* (pp. 1-4). IEEE.
- Daskalaki, S., & Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1), 106-120.
- Davoudzadeh, M., Rafeh, R., & Rashidi, R. (2009). A Linear Solution for the University Timetabling Problem. In *Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on* (Vol. 2, pp. 54-57). IEEE.
- Di Stefano, C., & Tettamanzi, A. G. (2001, April). An evolutionary algorithm for solving the school time-tabling problem. In *Workshops on Applications of Evolutionary Computation* (pp. 452-462). Springer, Berlin, Heidelberg.
- Ferdoushi, T., Das, P. K., & Akhand, M. A. H. (2014). Highly constrained university course scheduling using modified hybrid particle swarm optimization. In *Electrical Information and Communication Technology (EICT), 2013 International Conference on* (pp. 1-5). IEEE.

- Fred, N. (2017). Students' perception of quality of time resources in teacher education programmes in the Universities in Nakuru County, Kenya. *Mara Research Journal of Humanities & Social Sciences-ISSN: 2519-1489*, 1(1), 86-93.
- Ganguli, R., & Roy, S. (2017). A study on course timetable scheduling using Graph Coloring Approach. *International Journal of Computational and Applied Mathematics*, 12(2), 469-485.
- Goltz, H. J., & Matzke, D. (1999). University timetabling using constraint logic programming. In *International Symposium on Practical Aspects of Declarative Languages* (pp. 320-334). Springer, Berlin, Heidelberg.
- Han, L., & Kendall, G. (2003). An investigation of a tabu assisted hyper-heuristic genetic algorithm. In *Evolutionary Computation, 2003. The 2003 Congress on* (Vol. 3, pp. 2230-2237). IEEE.
- Han, L., & Kendall, G. (2003). Guided operators for a hyper-heuristic genetic algorithm. In *Australasian Joint Conference on Artificial Intelligence* (pp. 807-820). Springer, Berlin, Heidelberg.
- Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. London, England: MIT Press.
- Houhamdi, Z., Athamena, B., Abuzaineddin, R., & Muhairat, M. (2019). A Multi-Agent System for Course Timetable Generation. *TEM Journal*, 2019(1), 211-221.

- Junn, K. Y., Obit, J. H., & Alfred, R. (2017). A constraint programming approach to solving university course timetabling problem (UCTP). *Advanced Science Letters*, 23(11), 11023-11026.
- Junn, K. Y., Obit, J. H., & Alfred, R. (2017). The Study of Genetic Algorithm Approach to Solving University Course Timetabling Problem. In *International Conference on Computational Science and Technology* (pp. 454-463). Springer, Singapore.
- Kostuch, P. (2004). The university course timetabling problem with a three-phase approach. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 109-125). Springer, Berlin, Heidelberg.
- Lewis, R. (2008). A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1), 167-190.
- Lih, B. C., San Nah, S., & Bolhassan, N. A. (2015). A study on heuristic timetabling method for faculty course timetable problem. In *IT in Asia (CITA), 2015 9th International Conference on* (pp. 1-3). IEEE.
- Lindahl, M., Mason, A. J., Stidsen, T., & Sorensen, M. (2018). A strategic view of university timetabling. *European Journal of Operational Research*, 266(1), 35-45.
- Mauritsius, T. (2018). A three phase heuristic for examination timetabling problem. In *2018 International Conference on Information Management and Technology* (pp. 1-9). IEEE.
- McCollum, B. (2007). A perspective on bridging the gap between theory and practice in university timetabling. *Practice and Theory of Automated Timetabling VI*, 3-23.

- Meisels, A., & Kaplansky, E. (2002). Scheduling agents—distributed timetabling problems. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 166-177). Springer, Berlin, Heidelberg.
- Mirrazavi, S. K., Mardle, S. J., & Tamiz, M. (2003). A two-phase multiple objective approach to university timetabling utilising optimisation and evolutionary solution methodologies. *Journal of the Operational Research Society*, 54(11), 1155-1166.
- Murray, K., Muller, T., & Rudova, H. (2006). Modeling and solution of a complex university course timetabling problem. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 189-209). Springer, Berlin, Heidelberg.
- Padmini, M. V., & Athre, K. (2010). Efficient design of university timetable. In *2010 IEEE International Conference on Electro/Information Technology* (pp. 1-5). IEEE.
- Pandey, J., & Sharma, A. K. (2016). Survey on university timetabling problem. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on* (pp. 160-164). IEEE.
- Parera, S., Sukmana, H. T., & Wardhani, L. K. (2016). Application of genetic algorithm for class scheduling (Case study: Faculty of science and technology UIN Jakarta). In *Cyber and IT Service Management, International Conference on* (pp. 1-5). IEEE.
- Petrovic, S., & Burke, E. (2004). University timetabling. In *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, chapter 45.
- Pillay, N. (2016). A review of hyper-heuristics for educational timetabling. *Annals of Operations Research*, 239(1), 3-38.

- Rossi-Doria, O., & Paechter, B. (2004). A memetic algorithm for university course timetabling. In *Combinatorial Optimisation 2004 Book of Abstracts*, 56. Lancaster University Lancaster, UK.
- Sabar, N. R., Ayob, M., Qu, R., & Kendall, G. (2012). A graph coloring constructive hyper-heuristic for examination timetabling problems. *Applied Intelligence*, 37(1), 1-11.
- Santiago-Mozos, R., Salcedo-Sanz, S., DePrado-Cumplido, M., & Bousoño-Calzón, C. (2005). A two-phase heuristic evolutionary algorithm for personalizing course timetables: a case study in a Spanish university. *Computers & Operations Research*, 32(7), 1761-1776.
- Sarin, S. C., Wang, Y., & Varadarajan, A. (2010). A university-timetabling problem and its solution using Benders' partitioning—a case study. *Journal of Scheduling*, 13(2), 131-141.
- Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2), 87-127.
- Schimmelpfeng, K., & Helber, S. (2007). Application of a real-world university-course timetabling model solved by integer programming. *OR Spectrum*, 29(4), 783-803.
- Sigl, B., Golub, M., & Mornar, V. (2003). Solving timetable scheduling problem using genetic algorithms. In *Information Technology Interfaces, 2003. ITI 2003. Proceedings of the 25th International Conference on* (pp. 519-524). IEEE.

- Socha, K., Knowles, J., & Sampels, M. (2002). A max-min ant system for the university course timetabling problem. In *International Workshop on Ant Algorithms* (pp. 1-13). Springer, Berlin, Heidelberg.
- Socha, K., Sampels, M., & Manfrin, M. (2003). Ant algorithms for the university course timetabling problem with regard to the state-of-the-art. In *Workshops on Applications of Evolutionary Computation* (pp. 334-345). Springer, Berlin, Heidelberg.
- Thanh, N. D. (2007). Solving timetabling problem using genetic and heuristic algorithms. In *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on* (Vol. 3, pp. 472-477). IEEE.
- Wahid, J., & Hussin, N. M. (2016). Combination of graph heuristics in producing initial solution of curriculum based course timetabling problem. In *AIP Conference Proceedings* (Vol. 1761, No. 1, p. 020104). AIP Publishing.