



**Faculty of Computer Science and Information Technology**

**Improved Variable Neighbourhood Descent (VND) to Solve Universiti  
Malaysia Sarawak (UNIMAS) Course Timetabling Problem**

**Chen Mei Ching**

**Doctor of Philosophy  
2023**

Improved Variable Neighbourhood Descent (VND) to Solve Universiti  
Malaysia Sarawak (UNIMAS) Course Timetabling Problem

Chen Mei Ching

A thesis submitted

In fulfillment of the requirements for the degree of Doctor of Philosophy

(Artificial Intelligence)

Faculty of Computer Science and Information Technology

UNIVERSITI MALAYSIA SARAWAK

2023

## 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: Chen Mei Ching

Matric No.: 19010037

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date : 22 August 2023

## ACKNOWLEDGEMENT

I would like to sincerely thank everyone who has contributed to this thesis, whether directly or indirectly.

First of all, I would like to express my thanks to my family members for giving me the support and motivation to pursue my PhD. They gave me the strength to keep pursuing my research journey.

My sincere gratitude goes to my supervisor, AP Dr Sze San Nah, and my co-supervisors, Dr Goh Say Leng and AP Dr Chiew Kang Leng, for their guidance and support in completing this study. Their valuable knowledge has been the greatest input in helping me to conduct my research. They have sacrificed their valuable time to meet me and help me during my PhD journey by giving many suggestions to improve my research output.

I also would like to express my gratitude to my friends who are willing to become channels for me to release my stress. They are willing to sacrifice their time to listen to me and be one of the outlets for me to release my emotions.

I also wish to express my appreciation to the Faculty of Computer Science and Information Technology for providing the facilities needed to do my research. My sincere thanks to the Faculty of Computer Science and Information Technology administrator/timetable planner for providing the real-world dataset needed for my research.

My sincere gratitude to the Centre for Graduate Studies for the advice and support given during my period of study at Universiti Malaysia Sarawak. Finally, I would like to thank the

management of the Universiti Malaysia Sarawak for making it possible for me to complete my study here in Sarawak. Thank you all.

## ABSTRACT

Academic institutions face the timetabling problem every semester. The task of allocating lectures to limited timeslots and venues must fulfil certain constraints unique to each educational institution. This study investigated heuristic orderings and the variable neighbourhood descent approach to tackle the course timetabling problem at the Faculty of Computer Science and Information Technology (FCSIT), Universiti Malaysia Sarawak (UNIMAS) on the basis of the students. The objectives of the study were to formulate a mathematical model and improve a computational bounded heuristics-based solution to solve the course timetabling problem at the faculty. A two-stage heuristic algorithm is proposed. In stage 1, heuristic orderings were utilised to find a feasible solution using 31 timeslots instead of the 48 timeslots in the existing timetabling software. In stage 2, the variable neighbourhood descent approach with new neighbourhood structures was utilised to improve the quality of the solution. The improved algorithm was tested on real-world data instances (in semesters 1 and 2 of 2019/2020) at the FCSIT, UNIMAS. The results show that certain heuristic orderings (the largest degree or the combination of the largest degree and largest enrolment in descending order) are better than others in generating a feasible solution. In stage 2, the proposed algorithm with new neighbourhood structures managed to reduce the soft constraint violations for instances in semesters 1 and 2. Sensitivity analysis was performed on the proposed algorithm. The experimental results demonstrate the flexibility of the proposed algorithm in solving the university course timetabling problem (UCTTP) at the FCSIT.

**Keywords:** University course timetabling, optimisation, heuristic orderings, variable neighbourhood descent, perturbation

***Penambahbaikan Penurunan Pembolehkan Kejiranan Untuk Menyelesaikan  
Masalah Penjadualan Kursus Universiti Malaysia Sarawak (UNIMAS)***

***ABSTRAK***

*Institusi akademik berhadapan dengan masalah penjadualan pada setiap semester. Tugas memperuntukkan kuliah ke slot masa dan tempat yang terhad mesti memenuhi kekangan tertentu yang unik bagi setiap institusi pendidikan. Dalam kajian ini, kaedah penyusunan heuristik dan penurunan pembolehkan kejiranan dalam menangani masalah penjadualan kursus di Fakulti Sains Komputer dan Teknologi Maklumat (FSKTM), Universiti Malaysia Sarawak (UNIMAS) dengan berasaskan pelajar disiasat. Objektif kajian adalah untuk menghasilkan model matematik dan membangunkan satu kaedah heuristik dengan komputasi terhad untuk menyelesaikan masalah penjadualan kursus di fakulti. Heuristik dengan dua fasa telah dicadangkan. Pada fasa 1, penyusunan heuristik digunakan untuk mencari jadual waktu yang sah menggunakan 31 slot masa dan bukannya 48 slot masa seperti dalam perisian penjadualan yang sedia ada. Pada fasa 2, kaedah penurunan pembolehkan kejiranan dengan struktur kejiranan yang baru digunakan untuk meningkatkan kualiti jadual waktu. Algoritma yang dicadangkan diuji dengan menggunakan data sebenar iaitu semester 1 dan 2 tahun akademik 2019/2020 dari FSKTM, UNIMAS. Keputusan menunjukkan kaedah penyusunan heuristik (darjah terbesar dan kombinasi darjah terbesar dan pendaftaran terbesar) adalah lebih baik daripada yang lain dalam menjana jadual waktu yang sah. Pada fasa 2, struktur kejiranan yang baru berjaya meningkatkan kualiti jadual waktu untuk semester 1 dan 2. Analisis sensitiviti dilakukan pada algoritma yang dicadangkan. Keputusan eksperimen menunjukkan fleksibiliti algoritma yang dicadangkan dalam penyelesaian masalah penjadualan kursus universiti di FSKTM.*

**Kata kunci:** *Penjadualan kursus universiti, pengoptimuman, penyusunan heuristik, penurunan pembolehubah kejitiranan, gangguan*



## TABLE OF CONTENTS

	<b>Page</b>
<b>DECLARATION</b>	i
<b>ACKNOWLEDGEMENT</b>	ii
<b>ABSTRACT</b>	iv
<i><b>ABSTRAK</b></i>	v
<b>TABLE OF CONTENTS</b>	vii
<b>LIST OF TABLES</b>	xiv
<b>LIST OF FIGURES</b>	xvii
<b>LIST OF ABBREVIATIONS</b>	xix
<b>CHAPTER 1: INTRODUCTION</b>	20
1.1 Study Background	20
1.2 Problem Statement	22
1.3 Objectives	23
1.4 Research Scope	24
1.5 Research Methodology	25
1.6 Research Significance	27
1.7 Thesis Outline	27

<b>CHAPTER 2: LITERATURE REVIEW</b>	<b>30</b>
2.1 Introduction	30
2.2 Approaches/ Methodologies in Benchmark UCTTP	30
2.2.1 Operational Research (OR) Based Techniques	31
2.2.2 Single Solution-based Meta-heuristics	33
2.2.3 Population-based Meta-heuristics	36
2.2.4 Hyper-heuristic	40
2.2.5 Multi Criteria/ Objective Approaches	40
2.2.6 Hybrid Approaches	40
2.3 Approaches/ Methodologies in Real-world UCTTP	42
2.3.1 Operational Research (OR) Based Techniques	42
2.3.2 Single Solution-based Meta-heuristics	42
2.3.3 Population-based Meta-heuristics	44
2.3.4 Hyper-heuristic	45
2.3.5 Hybrid Approaches	45
2.4 Perspectives in UCTTP	48
2.5 Trends in Benchmark UCTTP	51
2.5.1 Socha Benchmark Dataset	53
2.5.2 ITC-02 Benchmark Dataset	53
2.5.3 ITC-07 (Track 2) Benchmark Dataset	54

2.5.4	ITC-07 (Track 3) Benchmark Dataset	55
2.5.5	Hard Benchmark Dataset	56
2.5.6	ITC-2019 Benchmark Dataset	58
2.6	Algorithm for the Approaches/methodologies in UCTTP	59
2.6.1	Variable Neighbourhood Descent	59
2.6.2	Tabu Search	62
2.6.3	Simulated Annealing	62
2.6.4	Genetic Algorithm	63
2.7	Limitations of the Approaches/methodologies in UCTTP	64
2.8	Research Opportunities in UCTTP	66
2.9	Research Gap Analysis	68
2.10	Conclusion	69
	<b>CHAPTER 3: CASE STUDY</b>	70
3.1	Introduction	70
3.2	Problem Description	70
3.2.1	Curriculum-based Course Timetabling	71
3.2.2	Timeslots	72
3.2.3	Venues	73
3.2.4	Pre-registration Data Practice	75
3.2.5	Size of Problem Instance	75

3.3	Modelling and Formulation	76
3.3.1	Entities	76
3.3.2	List of Hard Constraints and Soft Constraints	77
3.3.3	Notations	78
3.4	Conclusion	84
<b>CHAPTER 4: PROPOSED ALGORITHM: HEURISTIC ORDERINGS AND</b>		
<b>VARIABLE NEIGHBOURHOOD DESCENT</b>		85
4.1	Introduction	85
4.2	Framework for Two-stage Heuristic Algorithm	85
4.3	Stage 1: Heuristic Ordering (HO)	86
4.3.1	Algorithm Description	86
4.4	Stage 1: Heuristic Ordering (HO) with a Perturbation	88
4.4.1	Algorithm Description	89
4.5	Stage 1: Tabu Search (TS)	93
4.5.1	Algorithm Description	93
4.6	Stage 2: Variable Neighbourhood Descent (VND)	96
4.6.1	Algorithm Description	96
4.7	Stage 2: Variable Neighbourhood Descent (VND) with a Perturbation	104
4.7.1	Algorithm Description	104
4.8	Stage 2: Tabu Search (TS) – Used as a Comparison to the Improved VND	105

4.8.1	Algorithm Description	106
4.9	Conclusion	108
<b>CHAPTER 5: REAL-WORLD UNIVERSITY COURSE TIMETABLING</b>		
	<b>PROBLEM</b>	110
5.1	Introduction	110
5.2	Dataset Description	111
5.3	Stage 1: Heuristic Ordering (HO)	111
5.3.1	Experimental Results and Discussion	111
5.4	Stage 1: HO_P	113
5.4.1	Experimental Results and Discussion	113
5.5	Stage 2: Variable Neighbourhood Descent (VND)	114
5.5.1	Experimental Results and Discussion	114
5.6	Stage 2: Variable Neighbourhood Descent (VND) with a Perturbation	121
5.6.1	Experimental Results and Discussion	121
5.7	Sensitivity Analysis	125
5.7.1	Number of Events	125
5.7.2	Timetable Availability and Large Venue Resources Availability	132
5.7.3	Number of Students	136
5.8	Comparison Between VND_Proposed and Tabu Search (TS)	139
5.8.1	Experimental Results and Discussion	139

5.9	Conclusion	141
<b>CHAPTER 6: ITC-2007 (TRACK 2) BENCHMARK DATASET</b>		144
6.1	Introduction	144
6.2	Problem Description	144
6.3	Dataset Description	146
6.4	Stage 1: Heuristic Ordering with a Perturbation (HO_P)	147
6.4.1	Experimental Results and Discussion	147
6.5	Stage 1: Tabu Search (TS)	148
6.5.1	Experimental Results and Discussion	148
6.6	Stage 2: Variable Neighbourhood Descent (VND)	150
6.6.1	Experimental Results and Discussion	150
6.7	Stage 2: Variable Neighbourhood Descent (VND) with a Perturbation	152
6.7.1	Experimental Results and Discussion	152
6.8	Comparison Between VND_Proposed and Tabu Search (TS)	153
6.8.1	Experimental Results and Discussion	153
6.9	Conclusion	155
<b>CHAPTER 7: CONCLUSION AND RECOMMENDATIONS</b>		156
7.1	Introduction	156
7.2	Conclusion of Research	156
7.3	Research Contributions	160

7.4	Future Work	161
	<b>REFERENCES</b>	162
	<b>APPENDICES</b>	172

## LIST OF TABLES

		<b>Page</b>
Table 2.1	Approaches in Solving UCTTP (Benchmark and Real-world Problem Datasets)	48
Table 2.2	Summary of Approaches in UCTTP	51
Table 2.3	Benchmark Datasets and Their Respective State-Of-The-Art Methodologies	52
Table 2.4	The Features of the Socha Benchmark Dataset	53
Table 2.5	The Features of the ITC-02 Benchmark Dataset	54
Table 2.6	The Features of the ITC-07 (Track 2) Benchmark Dataset	55
Table 2.7	The Features of the ITC-07 (Track 3) Benchmark Dataset	56
Table 2.8	The Features of Hard Benchmark Dataset	57
Table 2.9	The Features of the ITC-2019 Benchmark Dataset	58
Table 3.1	FCSIT Data Instances for Academic Years 2019/2020	70
Table 3.2	Usage of Blocked Timeslots	73
Table 3.3	Teaching Venues and their Capacity	74
Table 3.4	Set Notations of the Mathematical Model	78
Table 3.5	Parameter Notations of the Mathematical Model	79
Table 5.1	FCSIT Data Instances for Academic Years 2019/2020	111
Table 5.2	Number of Unallocated Courses for Initial Solutions Generated using Different HO in Stage 1. $N = 30$ runs.	112
Table 5.3	Comparison of the Number of Unallocated Courses using HO and HO_P in Stage 1 for Semester 1's instance. $N = 30$ runs.	113
Table 5.4	Comparison of Number of Unallocated Courses using HO and HO_P in Stage 1 for Semester 2's instance. $N = 30$ runs.	114
Table 5.5	Comparison Between the Number of Soft Constraint Violations that occurred using VND_Proposed and the Existing FCSIT Solution. $N = 30$ runs.	117



Table 5.6	Summary of Soft Constraint Violations for the New NS (NS3 and NS7) in Stage 2 for both instances (semesters 1 and 2) and their Paired Samples <i>t</i> -test Results. <i>N</i> = 30 runs.	120
Table 5.7	Summary of Soft Constraint Violations for the New NS (NS4 and NS8) in Stage 2 for Both Instances (semesters 1 and 2) and their Paired Samples <i>t</i> -test Results. <i>N</i> = 30 runs.	121
Table 5.8	Comparison of Soft Constraint Violations between VND_Proposed and VND_Proposed_P in Stage 2 for Both Instances (Semesters 1 and 2) and their Paired Samples <i>t</i> -test Results. <i>N</i> = 30 runs.	124
Table 5.9	Comparison of Soft Constraint Violations between VND_Proposed, VND_Proposed_P and Existing FCSIT Solution. <i>N</i> = 30 runs.	125
Table 5.10	Feature of Two Extra Events Added to the Database for Semester 1's instance for Sensitivity Analysis	126
Table 5.11	Comparison of the Feature for Semester 1's Instance Before and After the Extra Two Events are Added	126
Table 5.12	Number of Unallocated Courses for Initial Solutions Generated Using Different HO in Stage 1 After Two Extra Events are Added to Semester 1's Instance. <i>N</i> = 30 runs.	127
Table 5.13	Comparison of Number of Unallocated Courses for Initial Solutions Generated Using HO and HO_P in Stage 1 for Semester 1's Instance After the Extra Two Events are Added. <i>N</i> = 30 Runs.	128
Table 5.14	Comparison of Number of Unallocated Courses for Semester 1's Instance (Original) and Semester 1's Instance (Extra Two Events) in Stage 1 for HO and HO_P. <i>N</i> = 30 runs.	129
Table 5.15	Feature of Five Extra Events Added to The Database for Semester 1's Instance for Sensitivity Analysis	129
Table 5.16	Comparison of the Feature for Semester 1's Instance Before and After the Extra Five Events are Added	130
Table 5.17	Number of Unallocated Courses for Initial Solutions Generated Using Different HO in Stage 1 After Five Extra Events are Added to Semester 1's Instance. <i>N</i> = 30 runs.	131
Table 5.18	Comparison of Number of Unallocated Courses for Initial Solutions Generated Using HO and HO_P in Stage 1 for Semester 1's Instance After the Extra Five Events are Added. <i>N</i> = 30 runs.	132
Table 5.19	Comparison of Number of Unallocated Courses for Semester 1's Instance (Original) and Semester 1's Instance (Extra Five Events) in Stage 1 for HO and HO_P. <i>N</i> = 30 runs.	132

Table 5.20	Comparison of Number of Unallocated Courses for Semester 1's Instance (Extra Five Events) in Stage 1 for HO and TS with 31 Timeslots. $N = 30$ runs.	133
Table 5.21	Comparison of Number of Unallocated Courses for Semester 1's Instance (Extra Five Events) in Stage 1 for HO When Available Timeslots are Increased from 31 Timeslots to 34 Timeslots. $N = 30$ runs.	135
Table 5.22	Comparison of Number of Unallocated Courses for Semester 1's Instance (Extra Five Events) in Stage 1 for HO_P When Available Timeslots are Increased from 31 Timeslots to 34 Timeslots. $N = 30$ runs.	135
Table 5.23	Comparison of the Feature for Semester 1's Instance Before and After the Extra Students (100 Students) are Added	137
Table 5.24	Number of Unallocated Courses for Initial Solutions Generated Using Different HO in Stage 1 After 100 Extra Students are Added to Semester 1's Instance. $N = 30$ runs.	138
Table 5.25	Comparison of Number of Unallocated Courses for Initial Solutions Generated Using HO and HO_P in Stage 1 for Semester 1's Instance After the Extra 100 Students are Added. $N = 30$ runs.	139
Table 5.26	Comparison of Number of Unallocated Courses for Semester 1's Instance (Original) and Semester 1's Instance (Extra 100 Students) in Stage 1 for HO and HO_P. $N = 30$ runs.	139
Table 6.1	Number of Timeslots Used in ITC-2007 (Track 2)	145
Table 6.2	The Features of the ITC-2007 (Track 2) Benchmark Dataset.	146
Table 6.3	Comparison of Number of Unallocated Courses using HO and HO_P in Stage 1 for ITC-2007 (Track 2) Instances. $N = 30$ runs.	147
Table 6.4	Comparison of Number of Unallocated Courses using HO_P and HO_P_TS in Stage 1 for ITC-2007 (Track 2) Instances. $N = 30$ runs.	149
Table 6.5	Comparison of Soft Constraint Violations using VND_5 6 and the Proposed VND_5 6 7 8 for ITC-2007 (Track 2) Instances. $N = 30$ runs.	151
Table 6.6	Comparison of Soft Constraint Violations between VND_5 6 7 8 and VND_5 6 7 8_P in Stage 2 for ITC-2007 (Track 2). $N = 30$ runs.	152
Table 6.7	Comparison of Soft Constraint Violations between VND_Proposed and Tabu Search . $N = 30$ runs.	154

## LIST OF FIGURES

	<b>Page</b>
Figure 2.1 Neighbourhood Structures	43
Figure 2.2 Case Studies of the Real-world UCTTP	47
Figure 2.3 Classification of Approaches in Benchmark and Real-world UCTTP	50
Figure 2.4 Variable Neighbourhood Descent Algorithm	61
Figure 2.5 Tabu Search Algorithm	62
Figure 2.6 Simulated Annealing Algorithm	63
Figure 2.7 Genetic Algorithm	64
Figure 3.1 Number of Timeslots	73
Figure 4.1 General framework for solving the UCTTP at the FCSIT, UNIMAS	86
Figure 4.2 HO Algorithm	88
Figure 4.3 HO Algorithm with a Perturbation	90
Figure 4.4 Perturbation Procedure for HO Algorithm	92
Figure 4.5 Perturbation Procedure for HO Algorithm in Diagram	93
Figure 4.6 TS Algorithm	95
Figure 4.7 Neighbourhood structures: NS1 to NS8	100
Figure 4.8 VND Algorithm Solving <i>SC1</i>	102
Figure 4.9 VND Algorithm Solving <i>SC2</i> and <i>SC3</i> and ITC-2007 (Track 2) Benchmark Datasets	103
Figure 4.10 VND with a Perturbation	104
Figure 4.11 Perturbation Procedure for the Proposed VND Algorithm	105
Figure 4.12 Stage 2 Tabu Search Algorithm	108
Figure 5.1 Comparison Between the Number of Soft Constraint Violations that occurred using VND_Literature and VND_Proposed for Semester 1's Instance. $N = 30$ runs.	115

Figure 5.2	Comparison Between the Number of Soft Constraint Violations that occurred using VND_Literature and the VND_Proposed for Semester 2's Instance. $N = 30$ runs.	116
Figure 5.3	Comparison of the Number of Soft Constraint Violations that Occurred Using VND_Literature, VND_1 2 3 5 6 7 and VND_1 2 4 5 6 8 for Semester 1's Instance. $N = 30$ runs.	118
Figure 5.4	Comparison of the Number of Soft Constraint Violations that Occurred Using VND_Literature, VND_1 2 3 5 6 7 and VND_1 2 4 5 6 8 for Semester 2's Instance. $N = 30$ runs.	119
Figure 5.5	Comparison of the Soft Constraint Violations that Occurred Using VND_Proposed and VND_Proposed_P for Semester 1's Instance. $N = 30$ runs.	122
Figure 5.6	Comparison of the Soft Constraint Violations that Occurred Using VND_Proposed and VND_Proposed_P for Semester 2's Instance. $N = 30$ runs.	123
Figure 5.7	Original Timeslots Availability and Timeslots Availability After Three Hours Timeslots are Added	134
Figure 5.8	Original Timeslots Availability and Timeslots Availability After 3 Timeslots are Added Using Reserved Timeslots	136
Figure 5.9	Comparison Between the Number of Soft Constraint Violations that occurred using VND_Proposed and Tabu Search for Semester 1's Instance. $N = 30$ runs.	140
Figure 5.10	Comparison Between the Number of Soft Constraint Violations that occurred using VND_Proposed and Tabu Search for Semester 2's Instance. $N = 30$ runs.	141

## **LIST OF ABBREVIATIONS**

UNIMAS	Universiti Malaysia Sarawak
FCSIT	Faculty of Computer Science and Information Technology

# CHAPTER 1

## INTRODUCTION

### 1.1 Study Background

Educational timetabling is defined as the task of allocating events such as exams, subjects and courses to rooms and timeslots by fulfilling certain constraints (Tan et al., 2021; Thepphakorn & Pongcharoen, 2020; Tan et al., 2020; Assi et al., 2018). Timetabling is a challenging combinatorial optimisation problem in theory and practice (Schaerf, 1999). Universiti Malaysia Sarawak (UNIMAS) devotes significant resources to the development of a feasible and high-quality course schedule for each faculty. Efficient course allocation may result in the more effective use of valuable resources such as utility resources (Burke et al., 2005). For example, reducing the number of timeslots needed to generate a feasible solution. This saves valuable resources, such as electricity costs when lectures can be avoided to be conducted at night time. Therefore, it is crucial to find an optimal configuration for the variables defined to achieve specific objectives (Habashi et al., 2018).

The university course timetabling problem (UCTTP) involves allocating a set of courses to limited resources - namely lecturers, venues and timeslots - by fulfilling certain constraints (Goh et al., 2020; Goh et al., 2019; Erdeniz & Felfernig, 2018; Goh et al., 2017). The UCTTP can be divided into two different categories based on the problem settings and requirements, namely the curriculum-based course timetabling problem (CBCTTP) and the post-enrolment course timetabling problem (PECTTP). The UCTTP at UNIMAS is closely related to the CBCTTP. Constraints can be classified into two types, hard and soft. The fulfilment of hard constraints is mandatory in generating a feasible timetable. For example, a student cannot attend two lectures at the same time, while a lecturer cannot lecture on more

than one course simultaneously. Meanwhile, the fulfilment of soft constraints is optional but will determine the quality of the timetable generated. For example, students should not have only one lecture hour per day and lecturers should not have to lecture after five pm.

The UCTTP is known to be NP-hard (Wang et al., 2019; Song et al., 2018; Assi et al., 2018; Gunawan, Ng, & Poh, 2012); that is, the problem cannot be solved exactly in polynomial time as the growth of the problem size and its complexity are exponential (Babaei et al., 2015; Bardadym, 1996). Exact algorithms are guaranteed to provide optimal solutions but they are applicable to small-sized problems (Schaerf, 1999). As an alternative, heuristic algorithms are often utilised to provide relatively good solutions in an acceptable time (Gora et al., 2010).

To date, many studies have been conducted on the UCTTP, tackling either benchmark or real-world UCTTPs. For most real-world search problems, automatically generating high-quality solutions is a difficult challenge (Muklason et al., 2019). Benchmark UCTTP are usually oversimplified and meant for objective comparison of methodologies. Meanwhile, real-world UCTTP focus on the practicality of solution at academic institutions. Even real-world UCTTP vary between them in terms of requirements due to different policies, education systems and cultures. The objective is to find a feasible timetable with the fewest possible soft constraint violations. Furthermore, the UCTTP requirements differ across academic institutions as policies and regulations are unique to each institution. This study addresses the UCTTP at the Faculty of Computer Science and Information Technology (FCSIT), UNIMAS, using a real-world dataset. The performance of the improved variable neighbourhood descent (VND) is investigated. Its performance is compared against the existing timetable, which had been constructed using commercial timetabling software.

## 1.2 Problem Statement

This research focuses on a real-world course timetabling problem at Universiti Malaysia Sarawak. The issues addressed are as follows:

### i. Pre-registration data practice

At UNIMAS, there is no practice of course pre-registration. Student course registration only takes place after the timetable has been generated. Therefore, the timetable generated cannot cater to all the students individually.

Student enrolment data is predicted by the program coordinators by gut feeling. There are uncertainties in preparing the student-course registration data. Eventually, the venue utilisation is low, especially since a large-capacity venue may be allocated but a low number of students register. Other than that, “critical courses” involving “graduating students” cannot be identified, creating drastic changes to the planned timetable. Critical courses refer to courses that must be registered by repeating students for them to graduate.

### ii. Curriculum-based course timetabling

Curriculum-based course timetabling means that the timetable generated is based on the curriculum plan for each program. This causes the “repeaters”, those who need to repeat the course/s they failed in the previous semester, to face a certain dilemma as they cannot attend both the repeated and current semester courses as these clash. The current timetable used by the FCSIT cannot cater to all the “repeater” students because each “repeater” is a unique case. For example, one student may need to repeat three courses from the previous semester, whereas another may only need to repeat one. In the worst case, a student may need to repeat courses from two or three previous semesters. The consequence may be that