A Classroom Time Table Management System Using Searching Techniques Emphasizing on A Flexible Scoring Scheme: A Case Study of Faculty of Engineering, Prince of Songkla University

Warakorn Sitthirit

A Thesis Submitted in Fulfillment of the Requirements for the Degree of Master of Engineering in Computer Engineering

Prince of Songkla University

2016

Thesis Title	A Classroom Time Table Management System Using Searching Techniques Emphasizing on A Flexible Scoring Scheme: A Case Study of Faculty of Engineering, Prince of Songkla University
Author	Mr. Warakorn Sitthirit
Major Program	Computer Engineering

Major Advisor

....

(Asst. Prof. Dr. Sangsuree Vasupongayya)

Examining Committee :

....Chairperson

(Asst. Prof. Dr. Pichaya Tandayya)

....Committee

(Asst. Prof. Dr. Sangsuree Vasupongayya)

....Committee

(Dr. Duenpen Kochakornjarupong)

The Graduate School, Prince of Songkla University, has approved this thesis as fulfillment of the requirements for the Master of Engineering Degree in Computer Engineering

....

(Assoc. Prof. Dr. Teerapol Srichana)

Dean of Graduate School

This is to certify that the work here submitted is the result of the candidate's own investigations. Due acknowledgement has been made of any assistance received.

....Signature
(Asst. Prof. Dr. Sangsuree Vasupongayya)
Major Advisor

....Signature
(Mr. Warakorn Sitthirit)
Candidate

I hereby certify that this work has not been accepted in substance for any degree, and is not being currently submitted in candidature for any degree.

....Signature

(Mr. Warakorn Sitthirit)

Candidate

Thesis Title A Classroom Time Table Management System Using Searching Techniques Emphasizing on A Flexible Scoring Scheme: A Case Study of Faculty of Engineering, Prince of Songkla University

Author Mr. Warakorn Sitthirit

Major Program Computer Engineering

Academic Year 2016

ABSTRACT

This thesis proposes a classroom timetable management system prototype which can respond to different user requirements. Depth-bounded Discrepancy search (DDS) is applied together with the ordering heuristic as a search algorithm of the system. In addition, hard and soft constraints are used as a guideline of the searching process to find a suitable solution. Objective models are used for scoring purpose. The user interface module is designed to allow the constraints to be defined by the users according to their requirements. The courses offered by Faculty of Engineering, Prince of Songkla University were used as the case study in this thesis. The results of the proposed prototype show the improvement over the original university schedules. The resulting schedules reduce the conflicts and increase the satisfaction of the user requirements. The experiments on ordering heuristics and constraints provide the effects of the configuration of the proposed system in order to create a guideline. The proposed system can be applied to other datasets with the assistant of the provided guideline.

Keywords: University course timetabling, Depth-bounded Discrepancy search, Heuristic, Objective model

ชื่อวิทยานิพนธ์	ระบบจัดตารางห้องเรียนโดยใช้เทคนิคการค้นหาเน้นหลักการให้คะแนนแบบยืดหยุ่น: กรณีศึกษาคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์
ผู้เขียน	นายวรากร สิทธิฤทธิ์
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ปีการศึกษา	2559

บทคัดย่อ

งานวิจัยชิ้นนี้มีวัตถุประสงค์เพื่อนำเสนอระบบต้นแบบสำหรับการจัดตารางเรียนตารางสอนเพื่อรองรับความต้องการที่หลากหลายของผู้ใช้ โดยใช้วิธีการค้นหาดิสเครเพนซีแบบมีขอบเขตความลึกหรือดีดีเอสถูกนำมาประยุกต์ใช้ร่วมกับการใช้วิทยาการศึกษาสำนึกจัดลำดับเพื่อช่วยในการค้นหาผลลัพธ์ นอกจากนี้ยังมีการใช้เงื่อนไขแบบแข็งและแบบอ่อนเพื่อช่วยในการแนะนำการค้นหาให้เป็นไปตามที่ผู้ใช้งานต้องการ แบบจำลองตามวัตถุประสงค์ได้ถูกนำมาใช้ในการพิจารณาการให้คะแนน ส่วนติดต่อผู้ใช้ถูกออกแบบขึ้นมาให้ผู้ใช้งานสามารถกำหนดเงื่อนไขเองได้เพื่อตอบสนองการความต้องการของตัวผู้ใช้งาน กรณีศึกษาเพื่อทดสอบระบบคือรายวิชาที่สอนในคณะวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์ ผลลัพธ์ที่ได้จากระบบต้นแบบมีประสิทธิภาพดีกว่าตารางเรียนตารางสอนต้นฉบับของมหาวิทยาลัย ผลลัพธ์แสดงให้เห็นว่าการละเมิดเงื่อนไขแบบแข็งลดลงเป็นจำนวนมาก และเพิ่มความพึงพอใจตามความต้องการของผู้ใช้มากขึ้น นอกจากนี้ระบบยังได้ถูกทดสอบด้วยเงื่อนไขที่มีความสำคัญแตกต่างกันและวิทยาการศึกษาสำนึกจัดลำดับที่หลากหลายเพื่อนำข้อมูลที่ได้มาสร้างเป็นแนวทางในการใช้งานระบบต้นแบบกับข้อมูลชุดอื่น

คำสำคัญ: การจัดตารางเรียนตารางสอนในระดับมหาวิทยาลัย, การค้นหาดิสเครเพนซีแบบมีขอบเขตความลึก, วิทยาการศึกษาสำนึก, แบบจำลองตามวัตถุประสงค์

ACKNOWLEDGEMENT

I would like to express my sincerest gratitude to my supervisor, Assistant Professor Dr. Sangsuree Vasupongayya, who gave me invaluable help throughout my thesis with her knowledge, motivation and patience. Her guidance helped me in all the time of research and writing of this thesis.

Beside my advisor, I would like to thank the other members of my committee, Assistant Professor Dr. Pichaya Tandayya and Dr. Duenpen Kochakornjarupong, for their suggestions and helpful comments. And, I am grateful to Faculty of Engineering, Prince of Songkla University who supporting data and facility to complete my thesis.

This thesis would have been impossible without the financial support of Prince of Songkla University under ENG550405S-0 and PRPM ID 12034.

I must also acknowledge groups of fellow students in Department of Computer Engineering for our discussions, exchanges of knowledge, skills and wonderful experiences during my graduate program.

Last but not the least, I would like to thank my parents, my family for the support they provided me for my entire life and my friends for their encouragements and friendships

Warakorn Sitthirit

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

CB-CTT	Curriculum-based Course Timetabling
DDS	Depth-bounded Discrepancy Search
UA	University Original Timetabling
SMO	Simple Mixed Objective
MO	Mixed Objective
OH	Ordering Heuristic
CH	Dataset characteristic
TD	Testing Dataset
HC	Hard Constraint
SC	Soft Constraint
UI	User Interface
OOP	Java Virtual Machine
JSON	JavaScript Object Notation
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
Eng	Engineering
ME	Mechanical Engineering
MtE	Mechatronic Engineering
CoE	Computer Engineering
EE	Electrical Engineering
BME	Bio Medical Engineering
IE	Industrial Engineering
MfE	Manufacturing Engineering
MnE	Mining Engineering
MaE	Materials Engineering
CE	Civil Engineering
EnE	Environmental Engineering

ChE	Chemical Engineering
MELab	Mechanical Engineering Laboratory Room
EELab	Electrical Engineering Laboratory Room
IELab	Industrial Engineering Laboratory Room
MnELab	Mining Engineering Laboratory Room
CELab	Civil Engineering Laboratory Room
ChE	Chemical Engineering Laboratory Room

CHAPTER 1
INTRODUCTION

1.1 Motivation

University course timetabling is classified as a resource allocation problem [1]. Course timetabling is a task to allocate available rooms and timeslots to each class in each semester of each academic year. The allocation is based on the constraints which are defined by the scheduler. However, the requirements in each institute may be slightly different depending on the resources and specific conditions such as the class should be assigned a timeslot that avoids the weekly faculty meeting or the scheduler has to spare available timeslots for the teacher who has an administrative work [2] [3] or, some institutes allow evening class. Therefore, the constraints must be adapted to fit with the institute requirements.

The requirement of the university course timetabling is an interesting and complex problem because the requirement can be tracked down to the department or curriculum level [4] [5]. For example, at Prince of Songkla University, the lecture classes from Faculty of Science, Faculty of Engineering and Faculty of Agro-Industry prefer the morning 1-hour timeslot. However, the lecture classes from Faculty of Management Sciences and Faculty of Economics prefer the afternoon 1.5-hour timeslot [4]. Some curriculums contain a lot of laboratory style courses while some curriculums contain a lot of small-size lecture-style classes or large-size lecture-style classes. Each curriculum places some restrictions on the room/equipment usages while some restrictions can be changed for each semester. Therefore, a course timetabling system that can adapt to the requirements, constraints and changes will be useful.

For example, the Faculty of Engineering at Prince of Songkla University offers 12 undergraduate degrees in engineering fields. There are total of 143 teachers, 674 classes from 202 subjects and several 57 rooms. The teachers are grouped into 7 departments. The rooms are spaced in different buildings. The subjects include 93 laboratory-type classes and 581 lecture-type classes. Thus, the scheduler has to allocate rooms and timeslots for a lot of classes in each semester. However, human errors can occur because of mistakes in managing big and complex data. Not only the size of the data that can cause a problem, but also the differences in data characteristics increases the complexity of the process. For example, the electrical laboratory classes are required in many curriculums. Such classes will link to many students and require multiple teachers.

To demonstrate the issues further, Table 1-1 shows 94 teacher time conflict observed from the original university schedules. Moreover, Table 1-2 shows 20 student time conflicts observed. And, Table 1-3 shows 29 room conflicts observed.

In summary, the original university schedule produces 143 conflicts which cause a lot of problems to all parties involved.

Meanwhile, student group timetables also have time conflicts. shows 20 conflicts in student group timetable. shows 29 time conflict in room timetables. Total time conflicts from timetable assigned by the registrar division are 143 conflicts causing a lot of problems when the timetables are used.

Table 1-1 Teacher time conflicts observed

Teacher	#Conflict	Details	
Sawit Tanthanuch	1	Fri 12.00-14.50	212-202 (03), 213-481 (01)
Kanadit Chetpattananondh	3	Mon 15.00-17.50	210-301 (02), 217-301 (01)
		Tue 12.00-14.50	210-301 (01), 211-231 (01)
		Tue 14.00-15.50	211-231 (01), 210-301 (02)
Pramote Jutaporn	1	Wed 15.00-17.50	217-405 (01), 212-202 (01)
Leang Khooburat	3	Mon 12.00-14.50	210-301 (01), 212-292 (01)
		Mon 15.00-17.50	210-301 (02), 217-301 (01)
		Tue 12.00-14.50	210-301 (01), 212-292 (01)
Sompat Roongtawanreongsri	1	Mon 12.00-14.50	210-301 (01), 210-211 (01)
Kiattisak Wongsopanakul	1	Mon 15.00-17.50	210-301 (02), 217-301 (01)
Booncharoen Wongkittisuksa	1	Mon 15.00-17.50	210-301 (02), 217-301 (01)
Phanumas Khumsat	1	Mon 12.00-14.50	210-301 (01), 210-406 (03)

Table 1-1 Teacher time conflicts observed (cont.)

Teacher	#Conflict	Details	
Anuwat Prasertsit	1	Wed 15.00-15.50	211-221 (01), 212-202 (01)
Pornchai Phukpattaranont	1	Fri 9.00-11.50	210-401 (01), 213-351 (01)
Phairote Wounchoom	1	Fri 9.00-11.50	210-401 (01), 210-251 (01)
Warit Wichakul	9	Mon 13.00-14.50	210-391 (03), 212-231 (01)
		Tue 15.00-17.50	210-391 (04), 212-231 (01)
		Tue 15.00-17.50	210-391 (04), 210-406 (04)
		Wed 14.00-16.50	210-391 (03), 210-391 (01)
		Wed 14.00-16.50	210-391 (03), 210-231 (01)
		Wed 15.00-16.50	210-391 (01), 210-231 (01)
		Fri 13.00-14.50	210-406 (01), 210-391 (04)
		Fri 13.00-14.50	210-406 (01), 210-391 (01)
		Fri 13.00-14.50	210-391 (04), 210-391 (01)
Sutham Niyomwas	1	Thu 13.00-14.50	237-203 (01), 216-392 (01)
Somkiat Nakgul	1	Mon 13.00-15.50	215-406 (01), 217-301 (01)
Prakit Honghirunruang	2	Mon 13.00-15.50	215-406 (01), 216-391 (01)
		Thu 13.00-15.50	216-392 (01), 216-391 (01)
Somchai Sae-ung	1	Thu 14.00-14.50	216-391 (01), 215-111 (05)
Passakorn Vessakosol	2	Mon 13.00-15.50	215-406 (01), 216-391 (01)
		Thu 14.00-14.50	216-391 (01), 216-212 (02)
Theerayut Leevijit	2	Wed 14.00-16.50	215-111 (03), 217-405 (01)
		Thu 13.00-15.50	216-392 (01), 215-407 (01)
Charoen Jaitwijitra	1	Fri 13.00-15.50	225-381 (01), 229-212 (03)
Sininart Chongkhong	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Paiboon Innachitra	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Juntima Chungsiriporn	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Pakamas Chetpattananondh	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Pornsiri Kaewpradit	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Suratsawadee Kungsanant	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Kulchanat Prasertsit	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Nattawan Klatkaew	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Ram Yamsaengsung	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)

Table 1-1 Teacher time conflicts observed (cont.)

Teacher	#Conflict	Details	
Sukritthira Ratanawilai	1	Fri 13.00-15.50	230-341 (01), 230-444 (01)
Lek Sikong	1	Fri 13.00-14.50	235-371 (01), 237-371 (01)
Prapas Muangjunburee	3	Mon 9.00-9.50	235-230 (02), 237-405 (01)
		Wed 11.00-11.50	235-230 (01), 237-407 (01)
		Fri 13.00-14.50	235-371 (01), 237-371 (01)
Weerawan Laosiripot	2	Tue 9.00-10.50	237-480 (01), 237-341 (01)
		Fri 13.00-14.50	235-371 (01), 237-371 (01)
Vishnu Rachpech	2	Tue 11.00-11.50	235-440 (01), 235-330 (01)
		Thu 8.00-9.50	235-440 (01), 235-320 (01)
Manoon Masniyom	1	Fri 13.00-14.50	235-371 (01), 237-371 (01)
Chanin Dumradkarn	5	Mon 13.00-15.50	235-320 (01), 237-380 (01)
		Tue 9.00-10.50	237-480 (01), 235-320 (01)
		Tue 9.00-10.50	237-480 (01), 200-101 (01)
		Tue 10.00-10.50	235-320 (01), 200-101 (01)
		Thu 8.30-9.50	235-320 (01), 200-101 (02)
Pitsanu Bunnaul	2	Tue 9.00-10.50	237-480 (01), 235-320 (01)
		Fri 13.00-14.50	235-371 (01), 237-371 (01)
Wikhanet Damkhong	1	Thu 8.00-9.50	235-440 (01), 235-320 (01)
Surapon Arrykul	1	Fri 13.00-14.50	235-371 (01), 237-371 (01)
Thawatchai Plookphol	1	Fri 13.00-14.50	235-371 (01), 237-371 (01)
Pairoj Kirirat	1	Mon 13.00-15.50	215-406 (01), 217-301 (01)
Danupon Tonnayopas	1	Fri 13.00-14.50	235-371 (01), 237-371 (01)
Kalayanee Kooptarnond	1	Fri 13.00-14.50	235-371 (01), 237-371 (01)
Woraphot Prachaseree	1	Tue 11.00-13.50	221-231 (01), 221-312 (01)
Nattha Jindapetch	7	Wed 14.00-16.50	210-391 (03), 210-391 (01)
		Wed 14.00-16.50	210-391 (03), 212-202 (01)
		Wed 15.00-16.50	210-391 (01), 212-202 (01)
		Thu 14.00-16.50	210-391 (02), 212-202 (02)
		Fri 12.00-14.50	212-202 (03), 210-391 (04)
		Fri 12.00-14.50	212-202 (03), 210-391 (01)
		Fri 13.00-14.50	210-391 (04), 210-391 (01)

Table 1-1 Teacher time conflicts observed (cont.)

Teacher	#Conflict	Details	
Krerkchai Thongnoo	2	Wed 14.00-16.50	210-391 (03), 210-391 (01)
		Fri 13.00-14.50	210-391 (04), 210-391 (01)
Pichet Trakarnsiri	2	Wed 9.00-11.50	226-435 (01), 226-302 (01)
		Fri 9.00-11.50	226-435 (02), 226-302 (01)
Supapan Chaiprapat	3	Tue 13.00-15.50	226-302 (01), 229-213 (02)
		Thu 13.00-15.50	226-401 (01), 229-213 (01)
		Thu 16.30-19.20	226-302 (03), 229-213 (03)
Chatchai Jantaraprim	4	Tue 14.00-16.50	242-201 (01), 241-202 (01)
		Wed 14.00-16.50	241-201 (01), 242-202 (01)
		Thu 14.00-16.50	242-201 (03), 242-202 (04)
		Fri 14.00-16.50	242-201 (04), 242-202 (03)
Suthon Sae-wong	1	Sat 13.30-15.20	242-207 (01), 242-207 (02)
Vitaya Mhadnui	4	Mon 14.00-16.50	215-111 (01), 215-111 (10)
		Tue 14.00-16.50	215-111 (07), 216-212 (01)
		Tue 15.00-17.50	216-212 (01), 215-111 (11)
		Wed 15.00-16.50	215-111 (10), 215-111 (11)
Chinnadit Songnam	1	Wed 13.00-15.50	215-201 (02), 215-111 (03)
Naret Jindapetch	1	Tue 13.00-15.50	215-201 (03), 215-111 (07)
Sanguan Tungbodhitham	2	Mon 13.00-14.50	229-211 (01), 229-212 (01)
		Mon 13.00-14.50	229-211 (01), 229-212 (01)

Table 1-2 Student time conflicts observed under the original schedule

Student Group	Period	Subjects
EngB	Tue 9.00-9.50	322-171 (03), 322-171 (04)
EngC	Tue 15.00-15.50	890-101 (13), 890-101 (14)
EngH	Tue 17.00-19.20	332-103 (02), 322-171 (06)
EngJ	Wed 17.00-19.20	332-103 (03), 322-171 (02)
EngK	Wed 17.00-19.20	332-103 (03), 322-171 (02)
2CoE(A)	Tue 10.00-10.50	241-207 (01), 242-208 (03)
2CoE(B)	Tue 9.00-9.50	242-205 (04), 242-206 (02)
2CoE(Phuket)(A)	Thu 8.00-8.50	223-253 (01), 242-212 (01)
2BME	Wed 15.00-17.50	212-202 (01), 210-231 (01)
2ME(B)	Thu 13.00-14.50	322-271 (04), 216-212 (02)
3EnE	Mon 13.00-13.50	221-414 (01), 223-322 (01)
3EnE	Tue 11.00-13.50	221-231 (01), 221-312 (01)
3EnE	Thu 8.00-10.50	221-342 (01), 223-321 (01)
3IE	Tue 13.00-15.50	225-346 (03), 225-347 (01)
3MfE(B)	Tue 13.00-15.50	225-346 (03), 226-313 (02)
4CoE(NW)	Mon 12.00-13.50	241-460 (01), 241-403 (01)
4MtE	Wed 11.00-12.20	216-435 (01), 211-433 (01)
4CE	Tue 13.00-15.50	221-482 (01), 221-312 (01)
4MnE	Fri 13.00-14.50	235-371 (01), 237-371 (01)
4MaE	Wed 9.00-11.50	226-435 (01), 237-407 (01)

Table 1-3 Room conflicts observed under the original schedule

Room	#Conflict	Details	
A201	1	Fri 11.00-11.50	216-391 (01), 215-221 (03)
A203	1	Thu 14.00-14.50	213-471 (01), 216-391 (01)
A403	1	Tue 9.00-9.50	210-431 (01), 212-251 (02)
S104	1	Thu 10.00-10.50	221-342 (01), 230-334 (01)
MELAB	4	Mon 13.00-14.50	216-303 (01), 216-406 (01)
		Mon 15.00-15.50	217-301 (01), 216-406 (01)
		Tue 13.00-14.50	216-303 (01), 216-406 (02)
		Thu 13.00-14.50	216-392 (01), 216-303 (01)
CE108	1	Wed 8.30-8.50	221-322 (01), 220-593 (01)
CELAB	7	Tue 13.00-13.50	221-231 (01), 224-211 (01)
		Tue 13.30-13.50	224-211 (01), 221-361 (01)
		Wed 13.00-15.50	221-341 (01), 221-452 (01)
		Wed 14.00-15.50	221-452 (01), 221-261 (02)
		Wed 14.00-15.50	221-341 (01), 221-261 (02)
		Thu 14.00-15.50	221-495 (01), 221-261 (01)
		Fri 13.00-15.50	221-324 (01), 223-323 (01)
CE109	1	Mon 10.00-10.50	221-451 (01), 220-503 (01)
IELAB	7	Mon 14.00-15.50	226-313 (01), 229-212 (01)
		Tue 13.00-15.50	226-302 (01), 226-313 (02)
		Tue 15.00-15.50	226-313 (02), 229-213 (02)
		Tue 15.00-15.50	226-302 (01), 229-213 (02)
		Wed 13.00-15.50	226-313 (03), 229-212 (02)
		Thu 13.00-15.50	226-313 (04), 229-213 (01)
		Thu 17.00-19.30	226-302 (03), 229-213 (03)
MnELab	5	Mon 13.00-14.30	235-320 (01), 236-411 (01)
		Tue 13.00-13.50	235-330 (01), 235-210 (01)
		Tue 13.00-13.50	235-330 (01), 237-201 (01)
		Tue 14.00-15.50	235-330 (02), 235-210 (01)
		Tue 14.00-15.50	235-330 (02), 237-201 (01)

This thesis focuses on proposing a system that utilizes a searching technique and the heuristics to guide the search into a good solution space. As a result, the proposed system can adapt to the changes of the requirements. Thus, the proposed system is more flexible to be applied on different datasets. To evaluate the proposed idea, the problem of scheduling courses for the Faculty of Engineering, Prince of Songkla University is used as a case study. The proposed system is developed and evaluated using the case study. The constraints and scoring schemes are created according to the requirements of Prince of Songkla University [4] [5]. DDS (Depth-bounded Discrepancy search) [6] is used as the core search engine of the proposed system in order to provide a suitable solution according to the required constraints and changes. The experimental results will be conducted on the prototype system which is developed in this thesis. Several different setting will be evaluated and studied in order to provide a guideline for applying the proposed system on different requirements.

The remaining of this chapter contains objectives and scope of work.

1.2 Objectives

1) To design and develop a prototype of a classroom timetabling application using a search technique together with the problem-specific heuristics.
2) To evaluate the prototype using the courses offered by the Faculty of Engineering, Prince of Songkla University.
3) To propose a general guideline for applying the prototype on other university course timetabling problems.

1.3 Scope of work

1) The workload in this study is focusing on all classes offered in the first semester of 2012 at Faculty of Engineering, Prince of Songkla University.
2) The prototype of the proposed system must be developed.
3) The prototype system accepts the requirements as input and produces a weekly schedule of all subjects.

- A requirement includes the room information, the subject information and the system setting information which is a set of parameters that can be adjusted.
- The output can be viewed in three formats including the timetables of each teacher, each group of students and each room.

4) Some free elective subjects offered in many curriculums are excluded from this work. However, such subjects can be considered as a unique scheduling problem. The guideline on how to resolve such unique problem is also provided in this work.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEWS

Background and literature reviews related to this thesis is given in this chapter as five sections. The first section provides an overview of a university course timetabling problem. Section 2.2 describes the process at Prince of Songkla University in order to create the timetable. Section 2.3 provides an overview of the constraints used in a university course timetabling problem. Section 2.4 presents DDS (Depth-bound Discrepancy search) which is applied as the core search engine of the proposed system. Section 2.5 describes the details of tools for developing the prototype in this thesis.

2.1 University course timetabling

University course timetabling can be classified into two groups [7]. The first one is post-enrollment based course timetabling, which is done after the student enrollment period. The second group is curriculum-based course timetabling (CB-CTT) [8], which is done before the student enrollment period. The case study in this thesis is classified as the second group. Four input information including the course information, timeslots, room information and curriculums, are required under a CB-CTT problem [8].

- Course information includes a set of sections, teachers, a set of classes, room type for each class, student capacity and student groups.
- The timeslot is a set of date and time available for scheduling classes.
- Room information includes the type of rooms, the number of rooms in each type and the room capacity.
- Curriculum for each student group describes which subject must be offered during the semester.

Currently, many researchers are still interested in the course timetabling problem. Mathematics models are applied in several methods including integer linear programming [8] [9] [10] [11] [12] and reduced graph coloring [13]. Logic programming [14] is also being interested. Recently, metaheuristics approaches are increasingly popular to solve the course timetabling problem [15]. The methods are categorized into two groups. The first group is called single solution metaheuristics including simulated annealing [16] [17] [18], iterative local search [19] and tabu search [20] [21] [22]. Another group is called population-based metaheuristics including genetic algorithm [23] [24] [25] [26], ant colony optimization algorithm [27], artificial bee colony algorithm [28], harmony search algorithm [29] and honey bee mating optimization algorithm[30].

Some approaches are combined as a hybrid method for improving the performance and reducing disadvantages of the algorithms including combination of mathematics model approach with metaheuristic approach [18], single solution metaheuristic approach with population-based metaheuristic approach [23] [28] [31] and both single solution metaheuristic approaches [32].

However, the performance of the algorithm depends on the characteristics of the dataset. The third international timetabling competition (ITC2011) [33] shows that no method dominates the others for all types of datasets. Some methods are better than another but they are also worse on some datasets. Thus, it is impossible to create a method that suits all datasets, because it depends on the institute rules, features, costs, and fixations [34]. To bridge the gap between theory and practice, several works focus on real-world implementation [35] [36] [21]. Some works create a guideline to measure or benchmark the results [34] [37]. McCollum [38] suggests that the implemented system requires an interface to assist the user in modeling the constraints.

This work will design and develop the prototype which will allow the requirements to be stated as a set of heuristics and constraints in order for the prototype to provide a suitable solution to the problem. The guideline on how to describe the requirements as the constraints is also given.

2.2 Prince of Songkla University course timetabling

The current process to generate a course timetabling at Prince of Songkla University [4] starts from the registrar division copying the last year timetable and forwarding it to the faculties for the verification purpose. If there is any change, the faculty staff will confirm the updated version data and send it back to the registrar division. After that, the registrar division assigns the timeslots and rooms for all changes and verify them before publishing the result on the university website. The sequence of the process is shown in Figure 2-1.

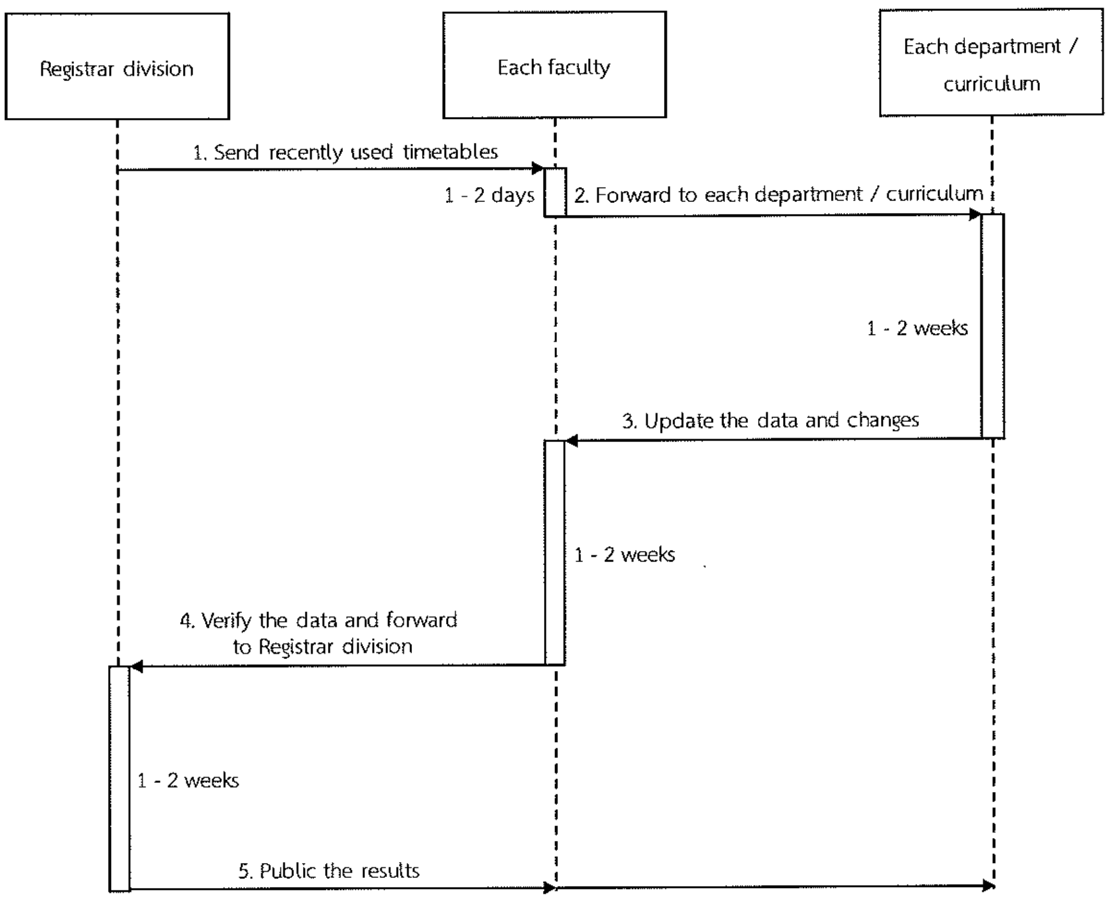


Figure 2-1 Sequence diagram of Prince of Songkla University timetable generating process

The guideline for timetabling at Prince of Songkla University is defined in the operating manual [4] as follows:

1) The timetable should be suitable with the faculty/department requirements.
2) The student timetable should be consistent with the teacher and room timetables.
3) The faculty/department has to change rooms or timeslots of the elective subjects instead of the main subjects.
4) The main subject timeslot should be similar to the last-year timetable.
5) The timetable should be harmonizing with the subject, timeslot, teacher and room.
6) The main subjects are more important than the elective subjects. The multiple-group of student subjects must be assigned rooms and timeslots before any single group of student subject.
7) Subject data includes study hours, subject type (lecture/laboratory) and teachers.
8) Student timetable should not contain any conflict according to the curriculum.
9) Teacher timetable should not contain any conflict according to their teaching assignment and other workloads, e.g. administrative works.
10) The room capacity must be large enough for each class.
11) Assigning the classes to their own department/faculty rooms first (if possible).
12) The classes should be assigned between 08:00 AM. and 04:00 PM., except for the specific cases.
13) Lunch break is from 12:00 PM. to 01:00 PM.
14) No class can be assigned on Saturday and Sunday timeslots except for the specific cases.
15) The science and technology subject should be assigned a three-hour lecture class on Monday, Wednesday and Friday using a one-hour timeslot type. The laboratory classes should be assigned in the afternoon or on Tuesday and Thursday morning.

The above 15 items will be used to create the initial constraints and heuristics in this thesis.

There are 12 majors in the Faculty of Engineering at Prince of Songkla University. However, this thesis will classify the workload into 66 student groups as shown in Table 2-1.

Table 2-1 Student groups in the dataset

Student Group	Description
Eng A	Group A of the freshman students in engineering curriculum
Eng B	Group B of the freshman students in engineering curriculum
Eng C	Group C of the freshman students in engineering curriculum
Eng D	Group D of the freshman students in engineering curriculum
Eng E	Group E of the freshman students in engineering curriculum
Eng F	Group F of the freshman students in engineering curriculum
Eng I	Group I of the freshman students in engineering curriculum
Eng J	Group J of the freshman students in engineering curriculum
Eng K	Group K of the freshman students in engineering curriculum
Eng L	Group L of the freshman students in engineering curriculum
Eng M	Group M of the freshman students in engineering curriculum
Eng N	Group N of the freshman students in engineering curriculum
Eng O	Group O of the freshman students in engineering curriculum
Eng P	Group P of the freshman students in engineering curriculum
Eng Q	Group Q of the freshman students in engineering curriculum
Eng R	Group R of the freshman students in engineering curriculum
2ME(A)	The first group of the sophomore students in mechanical engineering curriculum
2ME(B)	The second group of the sophomore students in mechanical engineering curriculum
2MTE	The sophomore students in mechatronic engineering curriculum
2CoE(A)	The first group of the sophomore students in computer engineering curriculum
2CoE(B)	The second group of the sophomore students in computer engineering curriculum
2CoE_PK(A)	The first group of the sophomore students in computer engineering Phuket campus curriculum

Table 2-1 Student groups in the dataset (cont.)

Student Group	Description
2CoE_PK(B)	The second group of the sophomore students in computer engineering Phuket campus curriculum
2EE	The sophomore students in electrical engineering curriculum
2BME	The sophomore students in bio medical engineering curriculum
2IE	The sophomore students in industrial engineering curriculum
2MfE	The sophomore students in manufacturing engineering curriculum
2MnE	The sophomore students in mining engineering curriculum
2MaE	The sophomore students in materials engineering curriculum
2CE	The sophomore students in civil engineering curriculum
2EnE	The sophomore students in environmental engineering curriculum
2ChE	The sophomore students in chemical engineering curriculum
3ME(A)	The first group of the junior students in mechanical engineering curriculum
3ME(B)	The second group of the junior students in mechanical engineering curriculum
3MTE	The junior students in mechatronic engineering curriculum
3CoE(A)	The first group of the junior students in computer engineering curriculum
3CoE(B)	The second group of the junior students in computer engineering curriculum
3EE (A)	The first group of The junior students in electrical engineering curriculum
3EE (B)	The second group of the junior students in electrical engineering curriculum
3BME	The junior students in bio medical engineering curriculum
3IE	The junior students in industrial engineering curriculum
3MfE (A)	The first group of The junior students in manufacturing engineering curriculum
3MfE (B)	The second group of The junior students in manufacturing engineering curriculum
3MnE	The junior students in mining engineering curriculum
3MaE	The junior students in materials engineering curriculum
3CE	The junior students in civil engineering curriculum
3EnE	The junior students in environmental engineering curriculum
3ChE	The junior students in chemical engineering curriculum
4ME(A)	The first group of the senior students in mechanical engineering curriculum
4ME(B)	The second group of the senior students in mechanical engineering curriculum
4MTE	The senior students in mechatronic engineering curriculum

Table 2-1 Student groups in the dataset (cont.)

Student Group	Description
4CoE(IT)	The senior students in the information technology branch of computer engineering curriculum
4CoE(HW)	The senior students in the computer hardware branch of computer engineering curriculum
4CoE(NW)	The senior students in the computer network branch of computer engineering curriculum
4CoE(RB)	The senior students in the robotics branch of computer engineering curriculum
4EE (Elect)	The senior students in the electronics branch of electrical engineering curriculum
4EE (Commu)	The senior students in the communication technology branch of electrical engineering curriculum
4EE (Power)	The senior students in the power electric branch of electrical engineering curriculum
4BME	The senior students in bio medical engineering curriculum
4IE	The senior students in industrial engineering curriculum
4MfE	The senior students in manufacturing engineering curriculum
4MnE	The senior students in mining engineering curriculum
4MaE	The senior students in materials engineering curriculum
4CE	The senior students in civil engineering curriculum
4EnE	The senior students in environmental engineering curriculum
4ChE	The senior students in chemical engineering curriculum

2.3 Constraint

The course timetable must be correct according to the requirements. The requirements can be viewed as a set of constraints. Thus, the constraints can be used for guiding the search to a suitable timetable. The review on constraints is given below.

2.3.1 Constraint category

The timetabling constraints can be categorized into 5 groups [15] including unary, binary, capacity, event-spread and agent. The structure of each group will be used for designing the prototype.

Unary constraint considers only one class or one group of classes. This constraint needs three inputs to create a relation between classes and resources. The relation puts the restriction or permission on how to use the resources. The structure of the unary constraint is shown in Figure 2-2.

Class	Relation	Resource
(Class/Group of classes)	(Allow/Disallow)	(Rooms/Timeslots)

Figure 2-2 Unary constraint structure

For example, the unary constraints can be

- Class A is allowed to be assigned in the morning timeslots.
- Class B is disallowed to be assigned the meeting rooms.

Examples of the unary constraints used in this work are

- Drawing classes are allowed to be allocated in evening timeslots.
- The computer laboratory classes must be in the computer laboratory rooms.

Binary constraint considers the relation between two classes. For example, a drawing lecture class has to take place before a drawing laboratory class.

Like, unary constraint, a binary constraint also needs three inputs. However, the relation between two classes is more complex than that of a single permission. An example of binary constraints in this work is “drawing lecture class has to take place before drawing laboratory class”. Figure 2-3 shows the structure of a binary constraint.

Class	Relation	Class
(Class/Group of classes)	Take place before/etc.	(Class/Group of classes)

Figure 2-3 Binary constraint structure

A capacity constraint considers the capacity of the room. That is the room must have enough seats to hold the class. The capacity constraint needs two inputs: class and the number of seats. An example of capacity constraint used in this work is “Classes that do not define the student capacity will be assigned to a room with 60 seats”. The structure of the capacity constraint is shown in Figure 2-4.

Class	Capacity
(Subject/ Group of subjects)	(Number of seats)

Figure 2-4 Capacity constraint structure

An event spread constraint considers the space between classes. This constraint will be spreading-out or clumping-together the classes depending on the user requirements. Event spread constraints can be controlled by the period of free timeslots between classes and the number of maximum total hours per day. If the users require more spreading-out timetable, they should increase the maximum free timeslots between classes or decrease the maximum total hours per day. On the other hand, if the users require a more clumping-together timetable, they have to decrease the maximum free timeslots between classes or increase the maximum total hours per day.

The input of event spread constraints is the maximum number of free timeslots between classes or the maximum total hours per day. However, the characteristic of each timetable can be different. Thus, the constraint has to specify which timetable will be affected. The structure of event spread constraints is shown in Figure 2-5.

Timetable (Student groups)	Parameter (Maximum number of free timeslots/ Maximum total hours per day)	Number (Timeslots)

Figure 2-5 Event spread constraint structure

For example, the event spread constraints used in this work are

- Maximum total hours per day of all students is 12 timeslots (6 hours).
- Maximum free timeslots of all students are 6 timeslots (3 hours).

An agent constraint considers the party in each timetable including teachers and student groups. The agent constraint gives a specific condition for some agents such as the relation between the agent and the resources. The agent constraint requires three inputs including agent, relation and resource. The relation will be given as a unary constraint on the resource. The structure of an agent constraint is shown in Figure 2-6.

Agent (Student groups/Teacher)	Relation (Allow/Disallow)	Resource (Rooms/Timeslots)

Figure 2-6 Agent constraint structure

For example, the agent constraints used in this work are

- The teacher of computer programming techniques class is allowed to be assigned to the Saturday morning timeslot.
- The graduate level classes are allowed to use the IDL meeting room.

2.3.2 Hard/soft constraints

As mentioned above that the constraints can be used for guiding the search in order to find a suitable solution. The type of the constraints can affect the result. Some constraints are more important than the others. For example, the constraint of assigning available rooms and timeslots for all classes according to the requirements is more important than the constraint of assigning the afternoon timeslots for the laboratory classes. Hence, some laboratory classes can be scheduled in the morning timeslots but all classes must be assigned the rooms and timeslots.

The violation of some constraints can cause problems. For example, if the constraint of assigning the available rooms for a class is violated, then two classes might be assigned to the same room. Thus, there are some constraints that cannot be violated which is called hard constraints. While the other type are called soft constraints. These are examples of hard constraints and soft constraints used in this work.

1) Hard constraints
    - The students must be able to enroll in all classes according to their curriculums.
    - The lecture class should be assigned the timeslot during the regular business hours.
    - The laboratory class should be finished before 6:00 PM.
    - No class should be assigned at noon timeslots.
2) Soft constraints
    - The same section class must not be assigned on the adjacent days.
    - The laboratory class should be assigned after 2:00 PM.

- The same section class should be assigned the same timeslots on each day.

The constraints in this work are created from the information provided by the registrar office [4].

2.3.3 Objective model

During the search in the solution space, the system must have an ability to evaluate the solution found in order to find a suitable solution. To evaluate the result, an objective model is used. The objective model is a method to compare two solutions in order to select a more suitable solution among solutions in the search space. To compare two results, the system requires a scoring technique to determine which result is better. The number of the constraint violations can represent the quality of the result. The less constraint violations result is better than the more one. Thus, the system requires a model to determine how the comparison the constraint. However, each constraint is different. For example, the hard constraints must dominate the soft constraints. Moreover, some soft constraints might be preferred over the others. Therefore, the objective model can help solving this issues. This thesis applies multi-objective model [39], including lexical, order-tradeoff and equal-tradeoff models.

- Lexical model can be represented as Lexical (A->B), where A and B are constraints. The model defines that A is more important than B. Thus, A will dominate B in all case.
- Ordered-tradeoff model can be represented as Tradeoff (A->B), where A and B are constraints. The model defines that A is more important than B. However, if the improvement of B is larger than the degradation of A, B which is less important can also win the competition.
- Equal-tradeoff model can be represented as Tradeoff (A: B), where A and B are constraints. The model defines that A and B are equal. Thus, the improvement and degradation of both constraints can affect the result equally.

2.4 Depth-bounded Discrepancy search

Depth-bounded Discrepancy search (DDS) [18] is a complete search technique. The search space can be viewed as a search tree. For explanation purpose, Figure 2-7 shows the search tree of eight solutions.

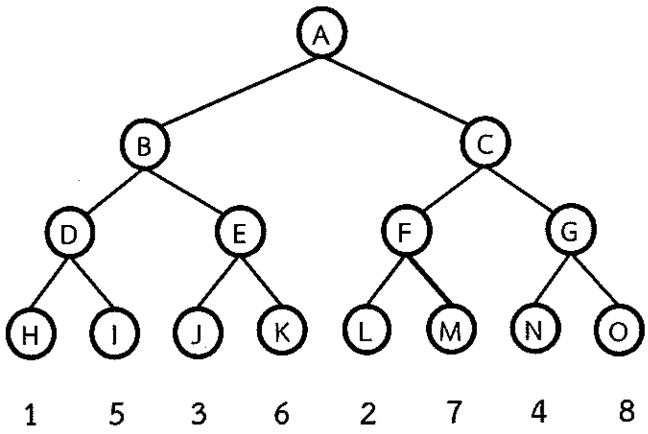


Figure 2-7 Depth-bounded Discrepancy search (DDS)

The search will start from the root to the leaf which is called a path or a solution. Once, the search reaches the leaf then a candidate solution is found. The objective model will guide the calculation of the quality of the solution. The search continues and the score will be calculated every time that the search reaches the leaf. If the new solution is better than the old solution, then the new solution is recorded. Furthermore, if the solution is no better than the current best solution then the path can be skipped in order to accelerate the search process. This effect is called pruning.

The progress of DDS will also be affected by the ordering of the branch at each level. That is, the search will start from the left most path (A, B, D, H) which is called the heuristic path. The heuristic is a guide of how to order the node at each level. For example, if each node is the class to be scheduled. The order of the classes to be considered by the scheduler is the heuristic. For example, "the laboratory-type class is considered before the lecture-type class" is one of the ordering heuristic.

The heuristic might be wrong in some cases and the effects of such incorrect heuristic can be monstrous. DDS prevents such gigantic effect by probing into each subtree in order to get a good baseline solution for pruning the search space. For example, Figure 2-7 shows the order of leaf to be discovered by the DDS. The first leaf is the right-most path (A, B, D, H). The next path is the 5th path (A, C, F, L) from the right which is a path from a different subtree. This property is suitable for the course timetabling problem because the earlier course assignments can have a huge impact on the later courses. Furthermore, the objective model can be calculated on a partial solution for the lexical model. Thus, the pruning technique can be used to avoid searching in the area containing solutions that are not better than the current solution found so far. In this work, DDS will be used as the main search technique in the proposed system.

2.5 Development tools

This section describes tools for implementing the proposed system which can be separated into three parts including the search engine, the database and the configuration.

The search engine is implemented using Java. Java [40] is an object oriented programming language (OOP) which executes processes on a virtual machine called Java virtual machine (JVM). Java is selected to create the search engine of the proposed system because the OOP scheme can be applied to use inheritance and polymorphism for the data model which can support the additional constraints, heuristics and agent later. The new search techniques can also be applied later. Java version 1.8.0 is used in this work.

MySQL [41] version 5.7.10 is used as the database of the proposed system. All datasets are in the table format. Thus, MySQL is a suitable choice.

CodeIgniter [42], AngularJS [43] and Bootstrap [44] are selected as tools for developing the remaining components of the proposed system. The user interface is required as the approach to configure the proposed system. The website helps the user to configure the system easily. because it can be accessed from a various web browser. The website is also separated into three parts.

The first part is CodeIgniter, which is a PHP framework. CodeIgniter purpose is to access the database and to prepare the data before forwarding it to the user interface. The AngularJS combined with Bootstrap is used for implementing the user interface. Moreover, CodeIgniter provides the web service which can be accessed from the various devices. CodeIgniter version 3.0.4 is applied for the proposed system.

AngularJS can receive JavaScript object notation (JSON) from the web service and bind the received data to the webpage in order to display the user interface. AngularJS version 1.2.16 is used for developing the proposed system.

Bootstrap is the tool for implementing the web interface from the template of hypertext markup language (HTML) and cascading style sheets (CSS). Bootstrap can be applied to bind the data from the web service with AngularJS and to display the web interface on the web browser simultaneously. Bootstrap version 3.3.6 with SB Admin version 2.0 template is used for creating the proposed system.

The website needs a web server to run the process. Apache [45] version 2.4.18 is used as the web server for running all parts of the website including CodeIgniter, AngularJS and Bootstrap.

CHAPTER 3
METHODOLOGY

This chapter presents the design of the system including the overview, search engine and scoring techniques. The workload and experimental setting are also given. Finally, the implementation details are given.

3.1 System overview

The course timetabling system requires the course data including classes, teachers and curriculums as the resources. The heuristics specify which dataset is chosen to create a search tree. The constraints are used as a decision to evaluate the results. The output of the system is the set of classes with rooms and timeslots that are assigned by the search engine. The input and output of the system are shown in Figure 3-1.

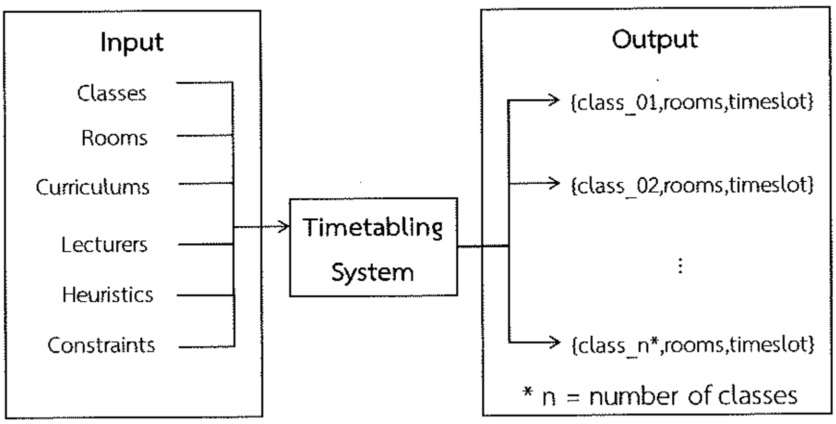


Figure 3-1 Input and output of the proposed system

The system uses Depth-bounded Discrepancy search (DDS) as the main search engine to assign the rooms and timeslots for each class. The structure of the system is shown in Figure 3-2. The search engine is the main method to search for a solution by evaluating each solution found so far using the scoring technique. The search engine first creates the search tree by using the timetabling and heuristic data. Section 3.2 will provide more information on how the search tree can be created.

Section 3.3 provides information on the scoring techniques and the constraints used. Section 3.4 provides the information regarding the workload characteristic. Section 3.5 gives the experimental setup. Finally, Section 3.6 gives the detail implementation of the prototype.

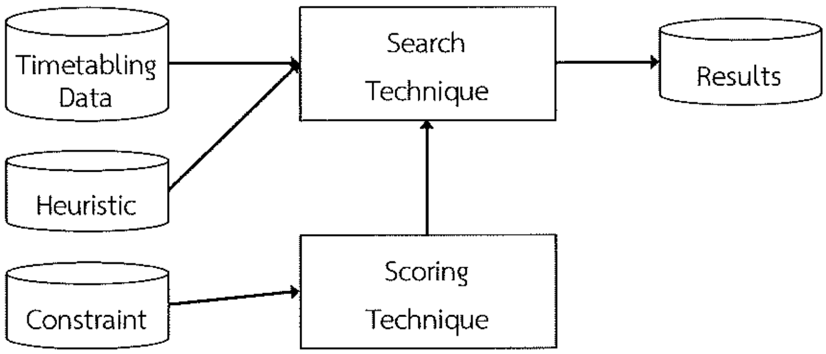


Figure 3-2 System structure

3.2 Search engine

The main search engine of the proposed system is DDS. The search engine requires a set of heuristics to decide how to build a search tree. Each node of the search tree represents a class as shown in Figure 3-3. Each path of the search tree represents a solution of the search engine. Thus, the whole search tree contains all possible solutions. The search engine will move from one solution to the next solution and keep the best solution found so far as the baseline to prune the search space as well. The heuristics which are defined by the scheduler will order the nodes in each path, where the left most path is the heuristic path. The other paths are the solutions that further away from the heuristic path. By this way, the length of all paths is equal.

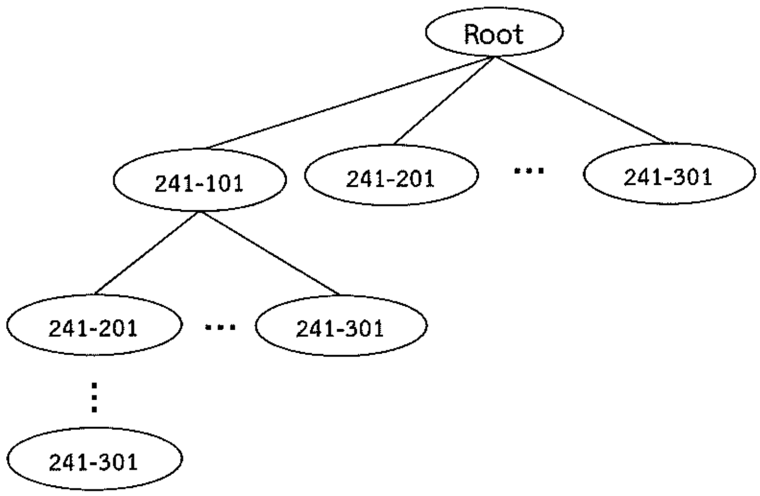


Figure 3-3 Search tree structure

The size of the search tree depends on the dataset. The scheduler can choose all classes in each semester to create the search tree or choose a partial data per search tree. However, the length of each path determines the length of the processing time for each result, because the larger search tree will result in the longer the search time. The necessary classes that should be in the same search tree are the classes that have a relation among themselves such as classes with mutual student groups or classes with mutual teachers. In this works, the search trees are created according to the student groups, because these students have mutual classes such as laboratories and main classes.

The order of nodes in each path is defined by the ordering heuristic. Figure 3-4 shows an example of an ordering heuristic which can be used for creating the search tree. The student group will be decided first. For each student group, the laboratory classes will be considered before main, elective and free elective classes, respectively. In the same classes type, the class hours will be used for ordering the classes. For example, a four-hour laboratory class will be considered before a three-hour laboratory class.

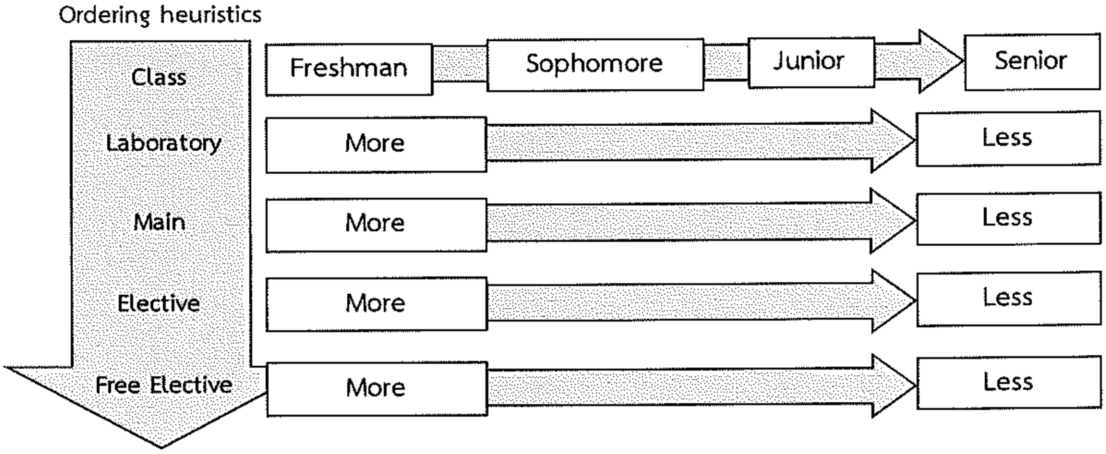


Figure 3-4 Example of an ordering heuristic for creating the search tree

Figure 3-5 shows the main process of the system which starts from connecting to the database in order to retrieve the data. The dataset and parameter must be set at this point. The selected dataset is then passed to the search engine. DDS searches the suitable rooms and timeslots for all classes in the dataset before the results are stored on the database. The process will be repeated until all datasets are processed.

In detail, the search engine searches through the search tree as shown in Figure 3-6. The search engine will move from path to path until the last path of the tree. The order of the paths to be discovered is decided by DDS. Each path will be evaluated. The path with the best score so far will be stored as the best solution found. The path with no good score or no better than the current best score will be cut in order to reduce the search time.

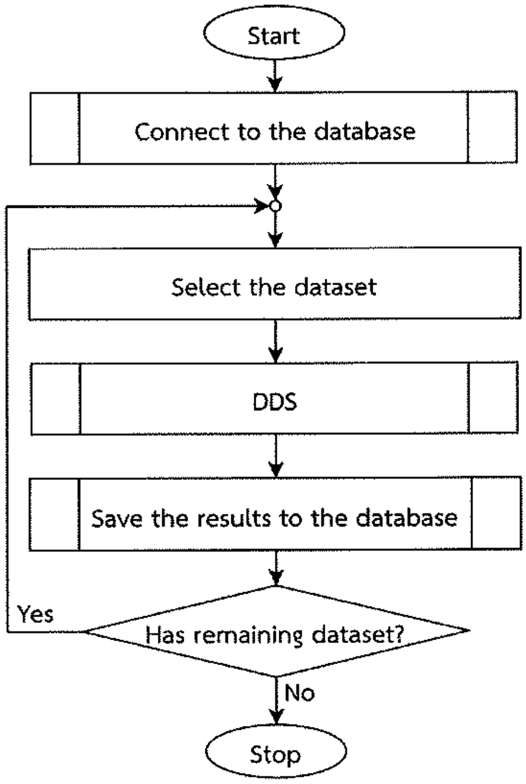


Figure 3-5 Flow chart of the main process in the system

To allocate the rooms and timeslots, the search engine builds the scoring table of all available timeslots in each node of the search tree. The process to create the scoring table is shown in Figure 3-7. The scoring engine has to check the availability of all related teachers, rooms and student group timeslots. Next, the scoring engine will calculate the constraint violations for all available timeslots, and return the score back to the search engine to create a scoring table. The search engine will compare all scores in the scoring table to choose the best timeslots and rooms. Next, the search goes along the path to get the result and compare the score of the result with the current best score found so far. The scoring technique is described next.

3.3 Scoring technique

The search engine requires the scoring technique in order to determine the score of the discovered solution during the search process. The simple method to calculate the score of the result is counting the number of constraint violations. However, each timetabling constraints are of different types, as described in Section 2. Thus, the engine requires a scheme to count the number of violations for each constraint type. The unary and binary constraints are counted by the number of violations that are occurred and multiplied by the number of student group timetables related to the class, in order to weight in the size of the effect.

Time-spread constraints are calculated differently. The number of timeslots represent the violation in this case. For example, the maximum timeslots per day constraint has more violations in more timeslots than that of the violation on the maximum timeslots per day. Thus, the number of timeslots that larger than the defined constraint limit will be counted and multiplied by the number of related agents, in order to weight in the size of the effect. However, the timeslot is counted as 30-minute slot. Thus, the number of timeslots must be divided by two in order to change the unit to hour.

Figure 3-7 and Figure 3-8 show the scoring algorithm. The process starts from creating a scoring table and checking the available timeslots. Next, the engine goes through the available timeslots and counts the constraint violations of the current timeslots and multiplying by the number of related student groups that are violated. The scoring engine will return the scoring table to the search engine as the information for making the final decision.

The score of each timeslot consists of the hard and soft constraint violations. The hard constraints are the constraints that should not be violated. If any hard constraint is violated, the result will be discarded. The comparison between two hard constraints (HC1, HC2) uses equal-tradeoff, Tradeoff (HC1: HC2), because the importance of all hard constraints are equal. However, some hard constraints are allowed to be violated. For example, the electrical engineering laboratory classes are allowed to allocated at noon timeslots due to the limited resources. The scoring engine will count all allowed hard constraint violations before the score will be stored in the scoring table.

As hard constraints dominate all soft constraints, the comparison between a hard constraint (HC) and a soft constraint (SC) is a lexical model, Lexical (HC-> SC). By this way, any soft constraint violation cannot overrule the effect of any hard constraint violation.

Unlike the hard constraint, the scheduler has to account for the importance of each soft constraint. The comparison among different soft constraints (SC1, SC2) is ordered-tradeoff model, Tradeoff (SC1-> SC2). The model defines that constraint SC1 is more important than constraint SC2. In case that the decreasing of SC2 violations is more than the increasing of SC1 violations, the SC2 which is less important can be preferred over SC1. The comparison among equal important soft constraints (SC3, SC4) is equal-tradeoff model, Tradeoff (SC3: SC4). Table 3-1, shows the hard constraints derived from the operation manual guideline in chapter 2 while Table 3-2 shows the soft constraints.

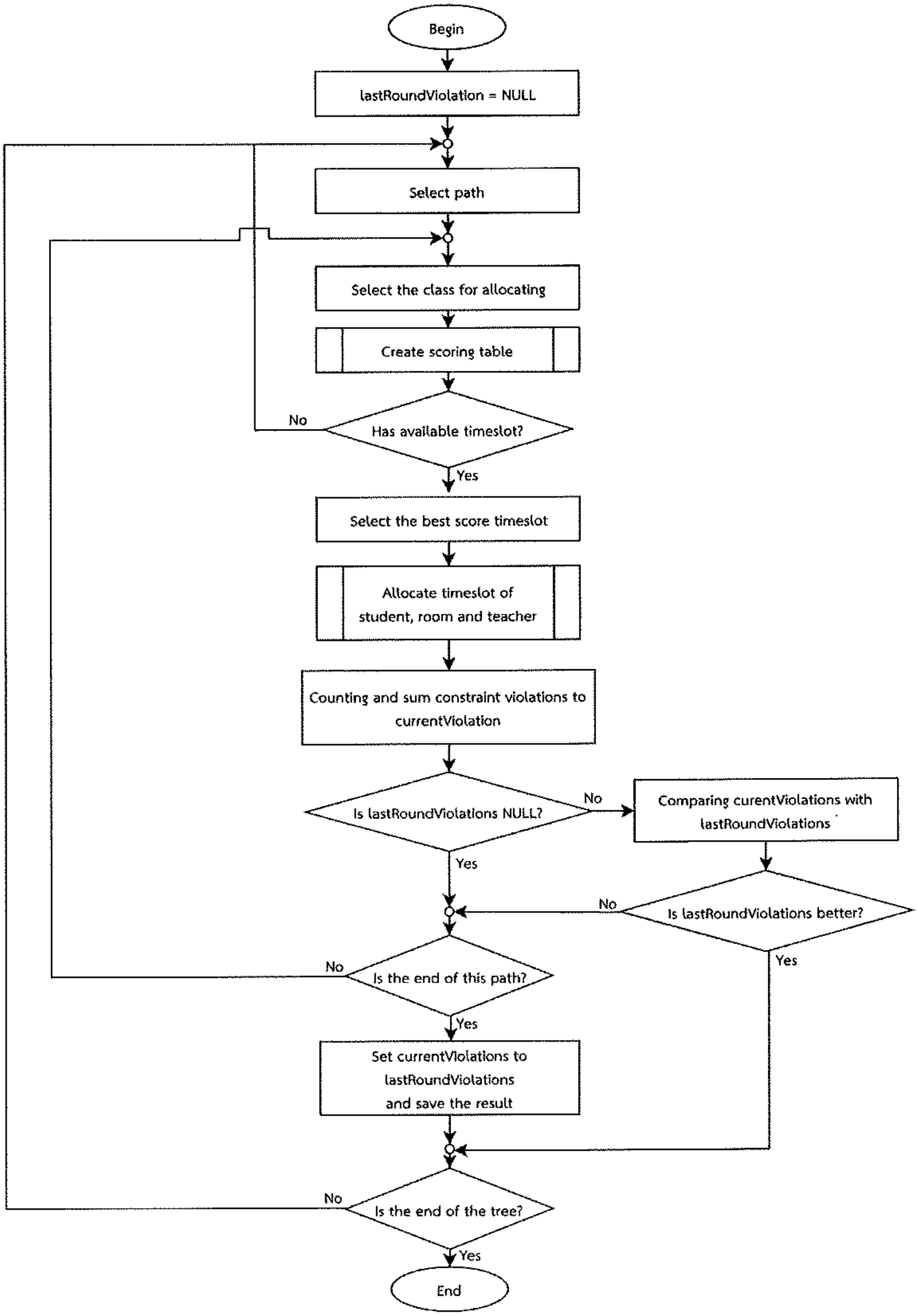


Figure 3-6 Main search flowchart

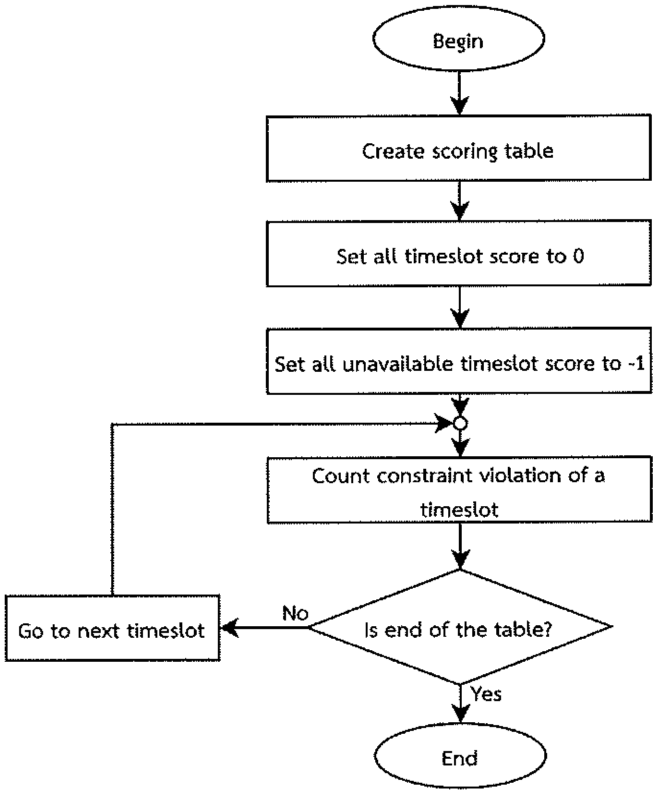


Figure 3-7 Scoring table creation flowchart

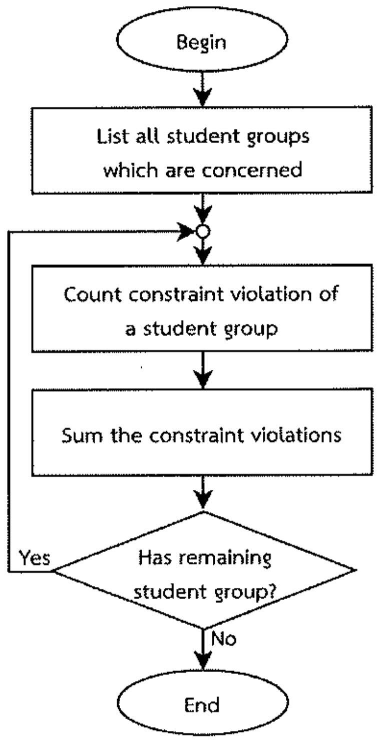


Figure 3-8 Timeslot scoring flowchart

Table 3-1 A list of hard constraints

Code	Description
H1	Student timetable conflict
H2	Lecture classes from the same section are not on the same day
H3	Teacher timetable conflict
H4	Lecture classes should be between 8.00 - 17.00
H5	Drawing classes and industrial engineering lab should be scheduled between 9.00 - 20.00
H6	Laboratory classes excepts industrial engineering lab should be scheduled between 9.00 - 18.00
H7	No classes is scheduled on Saturday and Sunday
H8	No classes at noon except electrical engineering lab
H9	No class before 8.00 and after 20.00
H10	Industrial engineering lab should be scheduled between 12.00 – 19.00
H11	Computer engineering lab should not be scheduled between 7.00 – 14.00

Table 3-2 A list of soft constraints

Code	Description
S1	Total hours per day must be less than 9 hours
S2	Free timeslot between classes must be less than 2 hours
S3	Classes from the same section should not be allocated to an adjacent day
S4	Computer engineering lab classes should be allocated after 14.00
S5	Classes from the same section should start at the same time on each day
S6	It is preferable that 3-hour classes are assigned in the afternoon
S7	No free timeslots between 1.5-hour classes
S8	Electrical engineering lab are to be assigned after 15.00 first
S9	Laboratory classes are to be assigned after 13.00 first
S10	Drawing classes should be allocated between 14.00-17.00 first

3.4 Workload

The workload in this thesis is a set of classes offered by the Faculty of Engineering at Prince of Songkla University during the first semester of 2012. There are 674 classes. The number of available rooms are 57, including laboratory rooms which are mechanical engineering (ME) laboratory room, electrical engineering (EE) laboratory room, computer laboratory rooms, hardware laboratory room, industrial engineering (IE) laboratory room, civil engineering (CE) laboratory room, mining engineering (MnE) laboratory room, chemical engineering (ChE) laboratory room and drawing room. There are 66 student groups as shown in Table 2-1. There are 7 departments and 12 curriculums. Table 3-3 shows the total class hours of students from each department. However, the first year students are set as a separated group. The first year students have the largest number of hours following by mechanical engineering students and computer engineering students.

To maintain a small tree size and allow the changes to parts of the problem, the tree is created according to the student groups. The freshman students are divided into two groups according to the current university setting. The sophomore, junior and senior students are divided into 7 groups according to their departments. This way, the students with shared classes are grouped together. The order of consideration within the same level is the total class hours as shown in Figure 3-9. Table 3-4 shows all 66 datasets.

The datasets consist of many different classes. Thus, the characteristics of the datasets can be different. Each dataset has its own characteristic. For example, the freshman timetables have a lot of study hours but most of them are pre-defined. Table 3-5 shows that 2ME timetable has similar numbers of 1-, 2- and 3-hour classes, while 3IE timetable contains more 3-hour classes. The number of teachers for each class is also considered, as shown in Table 3-7. Some datasets have less multiple teachers such as 2CoE, 2CE, 3CoE and 4CoE. Some datasets have many multiple teachers such as 3EE, 3MnE and 4MnE. Furthermore, the number of student groups for each class are shown in Table 3-7. Table 3-8 shows the teacher workload. Moreover, the total hours of each dataset and the room requirement are also considered. The following list of rules shows the characteristic of the datasets.

- If any group of classes in the dataset has a standard deviation (SD) larger than 0.7, that group is considered to have a different-class-size characteristic.
- If any group of classes has multiple teachers in more than 35% of the classes, that group is considered to have multiple-teacher characteristic.
- If any group of classes has a teacher with his/her difficulty factor larger than 42, that group is considered to have a high workload teacher characteristic. The difficulty factor of each teacher is calculated from the summation of the total-hour-per-week of each class-size square. This way a total hours of a large class-size will get a high weight.
- If any group of classes allow teacher-time-conflict, that group is considered to have teacher-time-conflict characteristic.
- If any group of classes has total hours larger than the median value of all dataset, that group is considered to have more-total-hour characteristic.
- If any group of classes has multiple student groups in more than 35% of the classes, that group is considered to have multiple-student-group characteristic.
- If there is an elective subject in the dataset, the dataset is considered to have elective-subject characteristic.
- If any classes in the dataset requires multiple rooms, that dataset is considered to have multiple-room characteristic.

The above rules are applied on the datasets presented in Table 3-4 in order to classify their characteristics into 9 groups as follows.

- CH1: Different-laboratory-class-size dataset includes 2ME, 2CoE, 2MnE, 3ME, 3IE, 3MnE, 3CE and 4CE.
- CH2: Different-lecture-class-size dataset includes Eng A - I, Eng J - R, 2EE, 2ChE, 3EE, 3IE,3MnE, 3CE, 3ChE, 4ME, 4EE, 4IE, 4MnE and 4 ChE.

- CH3: Many multiple-teachers dataset includes Eng A - I, Eng J - R, 2ME, 2EE, 2IE, 2MnE, 2ChE, 3EE, 3IE, 3MnE, 4ME and 4MnE.
- CH4: High-workload-teacher dataset includes Eng J - R, 2CoE, 2IE, 3IE, 3MnE, 3CE, 3ChE, 4ME, 4EE, 4IE, 4MnE and 4ChE.
- CH5: Teacher-time-conflict dataset includes 2ME, 2CoE, 2EE, 3ME, 3CoE, 3EE, 3IE, 3MnE, 3CE, 4EE, 4IE, 4MnE, 4CE and 4ChE.
- CH6: More total-hours dataset includes Eng A - I, Eng J - R, 2ME, 2CoE, 2IE, 3ME, 3EE, 3IE, 3MnE, 3CE, 4ME, 4MnE and 4CE.
- CH7: Many multiple-student-groups dataset includes 2IE, 2MnE, 2CE, 2ChE, 3ME, 3IE, 4ME.
- CH8: Elective-subject dataset includes 3MnE, 3CE, 4ME, 4CoE, 4EE, 4IE, 4MnE and 4CE.
- CH9: Multiple-room dataset includes 4ME.

Numbers of characteristics of a single dataset can be used as an indicator of the difficulty level to allocate the rooms and timeslots. For example, 3CE dataset which contains 6 characteristics has a high possibility to cause a problem during the scheduling process than 2EE dataset which contains only 2 characteristics. The experimental setting in this thesis aims to evaluate the effects of constraints and ordering heuristics on various dataset characteristics.

Table 3-3 Total hours of each department timetable.

Department	Total hours
1st year student	197.5
Mechanical engineering	176.5
Computer engineering	161.5
Electrical engineering	160
Industrial engineering	156
Mining engineering	140.5
Civil engineering	135.5
Chemical engineering	70.5

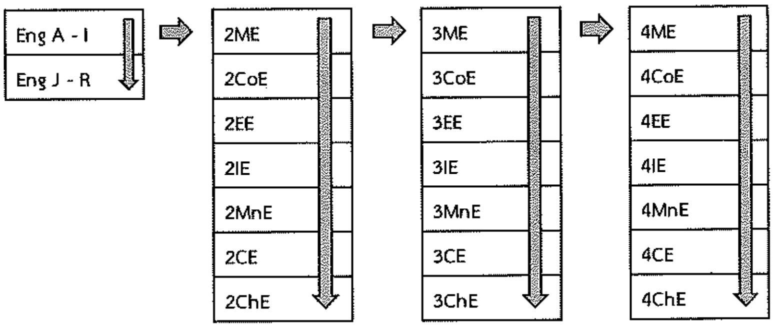


Figure 3-9 Order of the dataset consideration in this thesis

Table 3-4 Dataset definition

Year	Dataset	Student groups
1	Eng A - I	Eng A, Eng B, Eng C, Eng D, Eng E, Eng F, Eng I
	Eng J - R	Eng J, Eng K, Eng L, Eng M, Eng N, Eng O, Eng P, Eng Q, Eng R
2	2ME	2ME(A), 2ME(B), 2MTE
	2CoE	2CoE(A), 2CoE(B), 2CoE_PK(A), 2CoE_PK(B)
	2EE	2EE, 2BME
	2IE	2IE, 2MfE
	2MnE	2MnE, 2MaE
	2CE	2CE, 2EnE
	2ChE	2ChE
3	3ME	3ME(A), 3ME(B), 3MtE
	3CoE	3CoE(A), 3CoE(B)
	3EE	3EE(A), 3EE(B), 3BME
	3IE	3IE, 3MfE(A), 3MfE(B)
	3MnE	3MnE, 3MaE
	3CE	3CE, 3EnE
	3ChE	3ChE
4	4ME	4ME(A), 4ME(B), 4MtE
	4CoE	4CoE(IT), 4CoE(HW), 4CoE(NW), 4CoE(RB)
	4EE	4EE(Elec), 4EE(Commu), 4EE(Power), 4BME
	4IE	4IE, 4MfE
	4MnE	4MnE, 4MaE
	4CE	4CE, 4EnE
	4ChE	4ChE

Table 3-5 Total hours classified by the class size and student group

Dataset	Total hours	Class size (hours)					
		1	1.5	2	2.5	3	4
Eng A - I	114.5	50	9	34	12.5	9	0
Eng J - R	119	30	7.5	36	12.5	33	0
2ME	69	20	6	22	0	21	0
2CoE	94.5	27	1.5	36	0	30	0
2EE	43	17	3	16	0	3	4
2IE	51	19	6	14	0	12	0
2MnE	41	15	3	8	0	15	0
2CE	35	11	9	12	0	3	0
2ChE	26.5	11	1.5	8	0	6	0
3ME	49.5	14	13.5	16	0	6	0
3CoE	39	29	0	4	0	6	0
3EE	69	20	3	18	0	24	4
3IE	62	15	0	14	0	33	0
3MnE	51.5	14	1.5	4	0	24	8
3CE	60.5	24	6	4	2.5	24	0
3ChE	22	15	0	4	0	3	0
4ME	50	11	12	6	0	21	0
4CoE	32	19	3	10	0	0	0
4EE	48	14	3	22	0	9	0
4IE	36	6	3	2	0	21	4
4MnE	49	12	0	16	0	21	0
4CE	54	19	9	14	0	12	0
4ChE	23	5	0	6	0	12	0

Table 3-6 Total hours classified by the group of students

Dataset	Total hours	Number of student group (hours)				
		1	2	3	4	>4
Eng A - I	114.5	28	33	21	15	17.5
Eng J - R	119	26	35.5	26.5	8.5	22.5
2ME	69	47	7	7.5	3	4.5
2CoE	94.5	87	1	0	4.5	2
2EE	43	31	7.5	4.5	0	0
2IE	51	23	10	6	7.5	4.5
2MnE	41	14	15	4.5	7.5	0
2CE	35	20	7.5	3	0	4.5
2ChE	26.5	16	3	3	0	4.5
3ME	49.5	32.5	16	0	0	1
3CoE	39	30	9	0	0	0
3EE	69	66	3	0	0	0
3IE	62	36	20	6	0	0
3MnE	51.5	42.5	5	3	0	1
3CE	60.5	43.5	17	0	0	0
3ChE	22	22	0	0	0	0
4ME	50	23	18	9	0	0
4CoE	32	25	6	0	0	1
4EE	48	34	8	6	0	0
4IE	36	30	6	0	0	0
4MnE	49	41	5	3	0	0
4CE	54	42	12	0	0	0
4ChE	23	17	6	0	0	0

Table 3-7 Total hours classified by the number of teacher of the class

Dataset	Total hours	Number of teacher (hours)				
		1	2	3	4	>4
Eng A - I	114.5	74.5	4.5	9	24.5	2
Eng J - R	119	55	18.5	26.5	19	0
2ME	69	42.5	4.5	6	13	3
2CoE	94.5	93.5	0	0	0	1
2EE	43	28.5	4.5	0	0	10
2IE	51	29	9	6	4	3
2MnE	41	18.5	4.5	13	2	3
2CE	35	32	0	0	0	3
2ChE	26.5	16.5	3	0	1	6
3ME	49.5	38.5	3	5	0	3
3CoE	39	30	9	0	0	0
3EE	69	23	9	15	0	22
3IE	62	34	12	16	0	0
3MnE	51.5	13	20	10	0	8.5
3CE	60.5	49.5	11	0	0	0
3ChE	22	19	0	0	0	3
4ME	50	33	5	3	0	9
4CoE	32	32	0	0	0	0
4EE	48	39	6	0	0	3
4IE	36	30	0	0	6	0
4MnE	49	15	17	10	3	4
4CE	54	51	3	0	0	0
4ChE	23	17	0	0	0	6

Table 3-8 Top 20 teachers with a high difficulty factor value

Teacher	Group of students	Hours	Difficulty
Vitaya Mhadnui	2CoE(Phuket)(A), 2CoE(Phuket)(B), วศJ, วศN, วศO, 2ME(A)	36	90
Boonrueing Manasurakarn	3IE, 3MfE(A), 3MfE(B), 4IE, 4MaE	21	55
Supapan Chaiprapat	3MfE(A), 3MfE(B), 2IE, 2MfE, 4IE, 4MfE	19	49
Sakesun Suthummanon	3IE, 3MfE(A), 3CE, 3MfE(B), 3MaE, 4EE(Elec), 4EE(Commu), 4EE(Power), 4MaE, 4BME	18	46
Runchana Sinthavalai	3CE, 3MnE, 4EE(Elec), 4EE(Commu), 4EE(Power), 4ME(A), 4ME(B), 4IE, 4BME	15	45
Yodduang Pannara	3IE, 4ChE, 4MfE, วศA, วศB, วศC, วศD, วศE, วศF, วศG, วศH, วศI, 2CoE(Phuket)(A), 2CoE(Phuket)(B)	13	43
Klangduen Pochana	3IE, 3MfE(A), 3MfE(B), 4IE	19	42.5
Pakamas Chetpattananondh	4ChE, 3ChE	15.5	42.25
Sirikul Wisutmethangoon	2MaE, 3MnE	22	42
Pijit Pitsuwan	3MfE(A), 3MfE(B)	15	39
Chaisri Suksaroj	3EnE	19	38.5
Sangsuree Vasupongayya	3CoE(A), 3CoE(B)	21	33
Woraphot Prachasaree	3EnE, 4CE	17	32.5
Kulchanat Prasertsit	4ChE, 2ChE, 3ChE	13.5	32.25
Nattawan Klatkaew	3ChE, 4ChE, 2ChE	14	32
Winit Jungcharoentham	3MnE, 3EnE, 4CE	11	29
Sompat Roongtawanreongsri	2BME, 2EE, 3BME, 3EE(A), 3EE(B), 2ChE, 2MfE, 2MtE, 4EE(Elec), 4EE(Commu)	16	28
Vishnu RachPhet	3MnE, 4MnE, 4MaE	12	28
Pichet Trakarnsiri	3MfE(A), 3MfE(B), 4MaE, 2IE, 2MfE, 4IE, 4MfE	10	28
Sanguan Tungbodhitham	2IE, 2MnE, 2MaE, 2MfE	12	28

3.5 Experimental setting

There are three main experiments to be conducted in this work. The first experiment is the evaluation of the hard and soft constraints in guiding the search engine to a good portion of the search tree. The three objective models are evaluated. The first model, simple mixed objective (SMO) assigns an equal priority to all soft constraints. The second model, mixed objective 1 (MO1) assigns a high priority to time constraints (TC). The third model, mixed objective 2 (MO2) assigns a high priority to time-spread constraints (TSC). The constraints to evaluate the results are from Table 3-1 and Table 3-2. Time constraints used are S8, S9 and S10. Time-spread constraints used are S1, S2, S3, S4 and S5. Hard constraints used are H1, H3, H4, H5, H6, H7, H8, H9 and H10. The results from this experiment can give the inside into the effects of constraints and objective models.

The second set of experiments focuses on the effects of ordering heuristic on different dataset characteristics. The ordering heuristics include 6 alternative orders of class size, teacher workload and study-hour as shown in Table 3-9.

Table 3-9 Ordering heuristic definition

Heuristic index	Ordering heuristic
OH1	Class size > Teacher > Study-hour
OH2	Class size > Study-hour > Teacher
OH3	Teacher > Class size > Study-hour
OH4	Teacher > Study-hour > Class size
OH5	Study-hour > Class size > Teacher
OH6	Study-hour > Teacher > Class size

DDS creates the search tree according to the above ordering heuristic. The constraints from Table 3-1 and Table 3-2 are applied to the search engine with an equal priority soft constraint. HC is used for representing the number of hard constraint violations and SC is used for representing the number of soft constraint violations.

The last set of experiments focuses on the performance of the proposed prototype in comparison with the results of the original university timetabling. The comparison will focus on the result of the prototype from the first depth of DDS and the second depth of DDS in order to show the performance of the proposed system versus the time requirement to complete the task as well.

3.6 Implementation

The system contains three parts which are shown in Figure 3-10. The DDS and scoring techniques are implemented in Java language. The course information, ordering heuristics, constraints and results are stored in a MySQL database. Java uses Java database connectivity (JDBC) to connect to the database. The timetabling information, ordering heuristics and constraints can be changed via the user interface (UI). The UI part is implemented as a web-based format. CodeIgniter, which is a PHP Library is used for connecting to the database in order to parse the data into JavaScript object notation (JSON) before send it to AngularJS, which is another JavaScript library. AngularJS uses the received data to connect the data with the Bootstrap, which is a template of hypertext markup language (HTML) and cascading style sheets (CSS), before display the results on the browser.

The search engine is presented in Figure 3-11. The search engine starts its process by getting the datasets from the database. Then, the main process sends a dataset and a heuristic to the dataset factory in order to retrieve the class list. The main process forwards the class list to the DDS to begin the search. The DDS creates the timetabling data from all related data of classes in the class list. The timetabling data collects classes, student groups, teachers and rooms data in the map and the constraint factory creates the constraint list. The scope of each constraint defines by the agent factory. After all related data are ready, the DDS allocates timeslots and rooms for each class in the class list. The allocation gets all related data from the timetabling data and the score calculation creates the scoring data. The score calculation calculates the score of each cell of the scoring table by counting the constraint violations from the constraint list inside the timetabling data. After the scoring table is calculated, the allocation compares all remaining available timeslots to get the best-score timeslot and returns the result to the DDS. The DDS gathers the results and searches until a suitable result is found before the DDS moves to the next dataset. The results of each process is stored on the database for analysis purposes.

The entities in the database which is used by the system, and the relations among them are presented in Figure 3-12. The data in the database is separated into three groups: timetabling data, search engine data and search engine configurations. The timetabling data includes classes, teachers, rooms and student groups. The relation between teachers and classes is teaching. Curriculum defines the class list for each student groups. Room used is a relation between classes and room types.

The results from the search engine are stored in the class result. The search tree of each dataset is saved. The violation value is stored according to each student group. The search engine configurations are also stored. The constraint type is used for classifying the constraint groups and agent in order to define the scope of the constraint. Heuristic is defined by the dataset type to create the class list for DDS.

Figure 3-13, Figure 3-14 and Figure 3-15 show the design of the prototype UI for configuring the search engine. The heuristic list page is shown in Figure 3-13. The users can change the order of the dataset for the search engine. A new heuristic can also be added.

Figure 3-14 shows the constraint list page. The hard and soft constraints are separated into two sections. The hard constraints are shown before the soft constraints. The list has a description of each constraint and also has a button to enable or disable the constraint. The right side of each row has an expanding button to show more details. The users can edit the constraints. The constraint configuration page is shown in Figure 3-15, which is displayed differently depending on the selected constraint type. Figure 3-15 shows the teacher time conflict constraint configuration. The users have to define the constraint name, constraint description, hard/soft constraint, constraint priority and scope of the constraint. The flexible subject list contains a list of subjects whose constraints can be violated. The other constraints in the system include:

1) Time constraint is a constraint which is defined to limits the available timeslots.
2) Student-time-conflict constraint is a constraint which is used for checking the conflict in the student group timetable.
3) Teacher-time-conflict constraint is a constraint which is used for checking the conflict in the teacher timetable.
4) Max-free-time constraint is a constraint which limits the size of the free timeslots between classes in the student group timetable.
5) Max-total-hour constraint is a constraint which limits the total hours per day in the student group timetable.
6) Start-same-timeslot constraint is a constraint which defines a set of classes to be allocated in the same starting timeslot on different days.
7) Not-same-day constraint is a constraint which defines a set of classes to be allocated on different days.
8) Not-adjacent-day constraint is a constraint which defines a set of classes not to be allocated on the adjacent days.

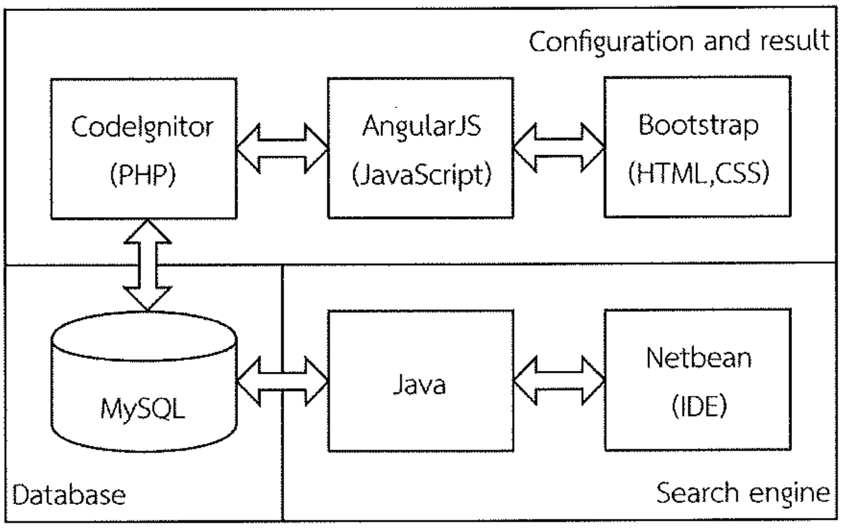


Figure 3-10 The proposed system structure

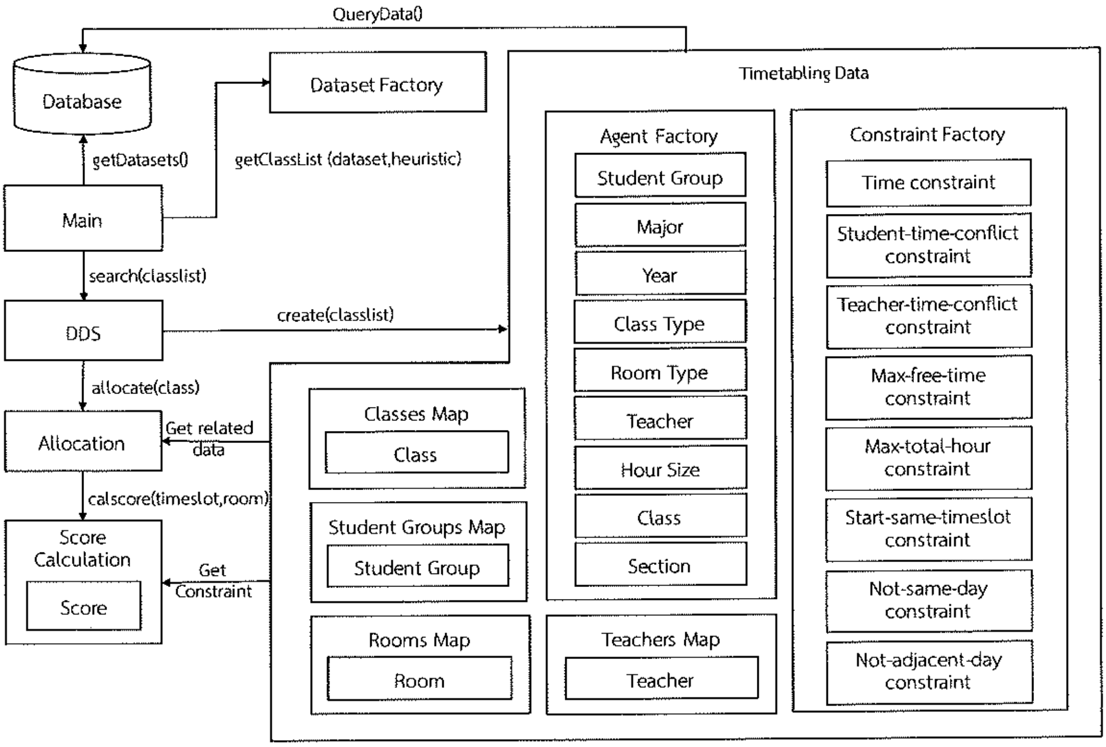


Figure 3-11 The search engine structure

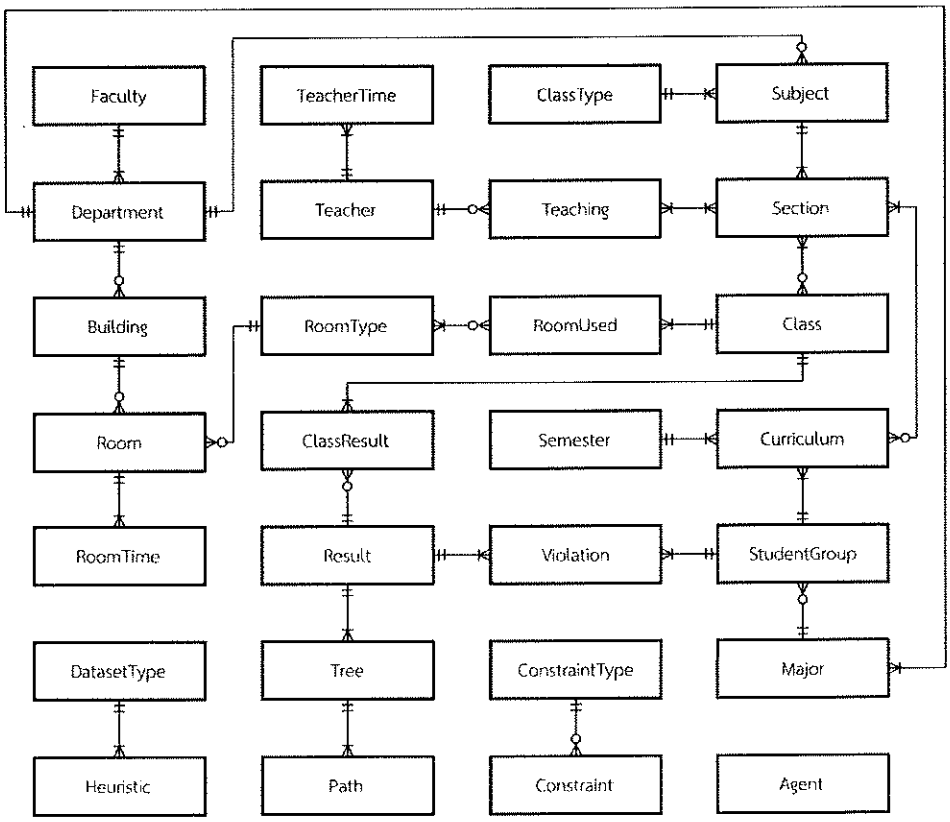


Figure 3-12 The database entity relations

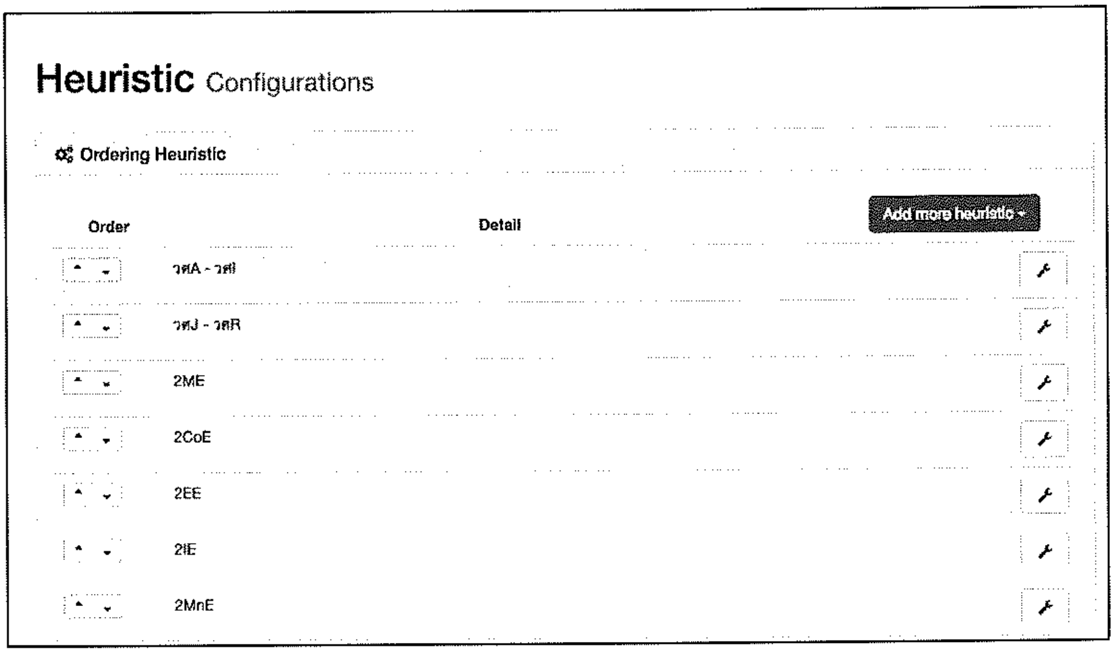


Figure 3-13 Heuristic configuration page

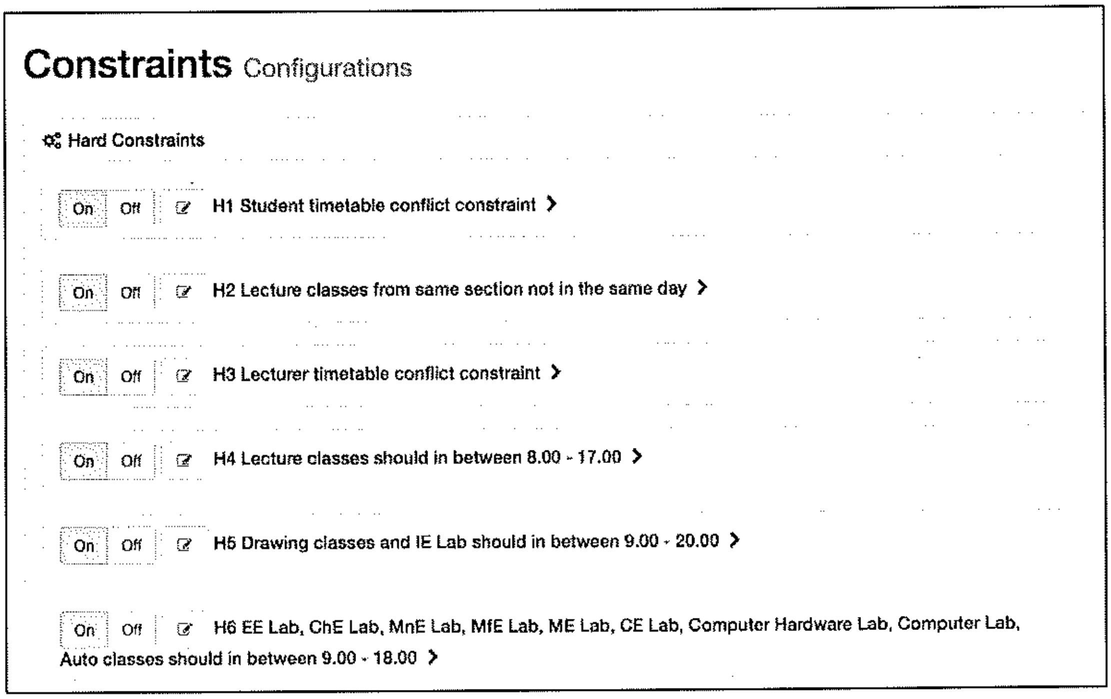


Figure 3-14 Constraint list page

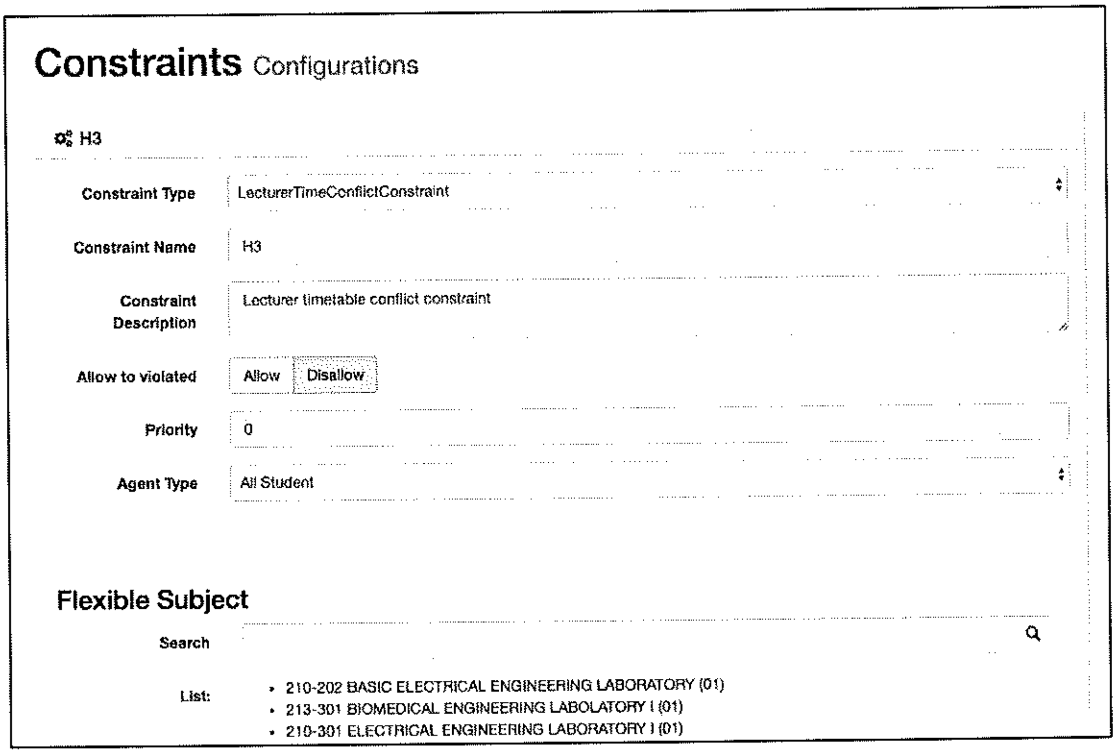


Figure 3-15 Constraint configuration page

The system consists of two parts. The first part is the search engine using Java language in the implementation. The second part is the configuration, which is shown in a web-based format. The database structure is designed according to the curriculum-based course timetabling data. The datasets from the case study are classified into nine characteristics which are described in Section 3.4. These datasets are used for studying the effects of each constraint and ordering heuristic.

CHAPTER 4
RESULTS AND DISCUSSIONS

The results and discussions are presented in this chapter. Section 4.1 shows the comparison results between the university original timetabling and the schedules from the proposed prototype. Section 4.2 to 4.4 show the results of the university original timetabling, the first DDS depth and the second DDS depth, respectively. Section 4.5 shows the effects of constraints and objective models. Section 4.6 shows the effects of the ordering heuristics. The performance evaluation of the search engine is shown in Section 4.7. The discussion on the results in order to provide a guideline for applying the proposed prototype on different datasets, is given in Section 4.8. The guideline is provided in Section 4.9.

4.1 Overall comparison results

This section provides the overall comparison results of the university original timetabling (UA) and the results of the proposed prototype.

Table 4-1 presents the comparison results on the number of hard constraint violations. The number of violations show that DDS can reduce a lot of hard constraint violations in comparison with that of the UA results. The first depth DDS result can reduce 75% of hard constraint violations, while the second depth DDS reduces 10% of hard constraints in comparisons with that of the first depth DDS.

Furthermore, the number of soft constraint violations of the DDS results is also reduced in comparison with that of the UA results. Table 4-2 presents the comparison results between the UA and the DDS results. The first depth DDS reduces 35% of soft constraint violations in comparison with that of the UA results, while, the second depth DDS reduces a few soft constraint violations further.

Figure 4-1, Figure 4-2 and Figure 4-3 present the timetabling results of junior environment engineering (3EnE) timetable of the UA, the first depth DDS and the second depth DDS results, respectively. The UA result shows a lot of student time conflicts in the timetable. The first and the second depth DDS results show no student time conflict on their results.

Moreover, the teacher timetables also have time conflicts. Figure 4-4, Figure 4-5 and Figure 4-6 present timetable of Asst.Prof. Leang Khooburat in the UA, the first depth DDS and the second depth DDS results, respectively. The UA timetable has a lot of time conflict. The first and second depth DDS results also show the time conflict. However, the second depth DDS can reduce the time conflict on the teacher timetable better than that of the first depth DDS.

Table 4-1 Hard constraint violations comparison between UA and DDS

Results	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	Total
UA	24	21	250	47	1	1	4	43	4	3	1	399
1st Depth	0	2	94	1	0	0	0	0	0	3	0	100
2nd Depth	0	0	87	0	0	0	0	3	0	0	0	90

Table 4-2 soft constraint violations comparison between UA and DDS

Results	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
UA	43	230	233	1	291	0	0	9	12	2	821
1st Depth	20	105	168	0	220	0	0	8	9	3	533
2nd Depth	3	118	135	0	227	0	0	7	13	3	506

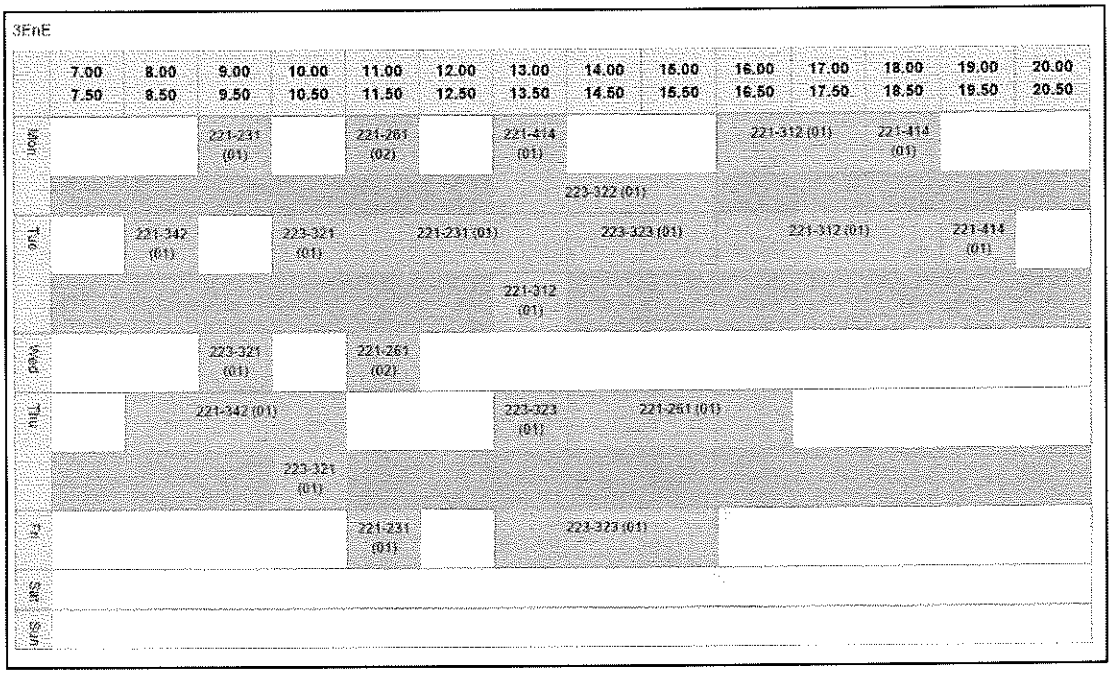


Figure 4-1 3EnE timetable of the UA results

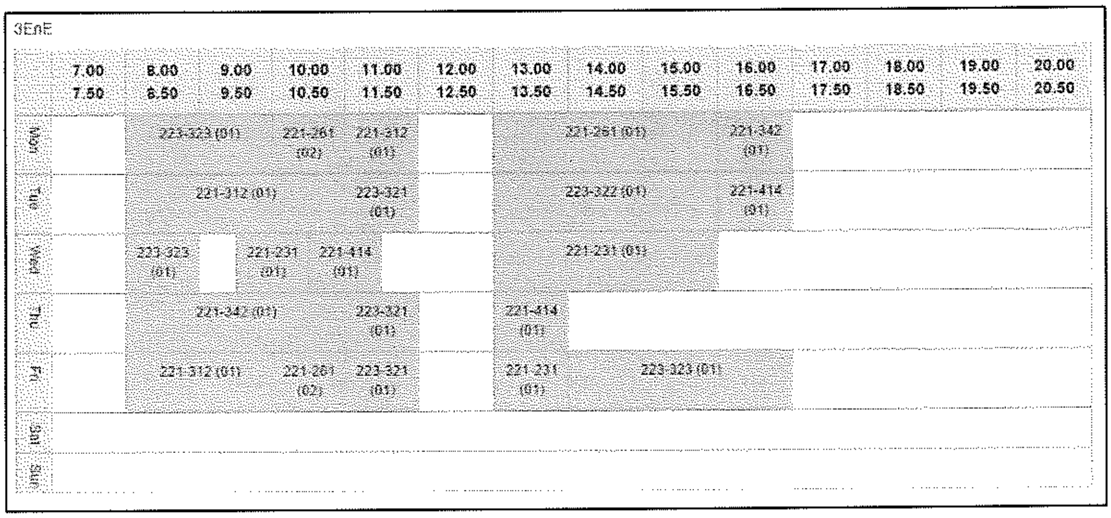


Figure 4-2 3EnE timetable of the first depth DDS results

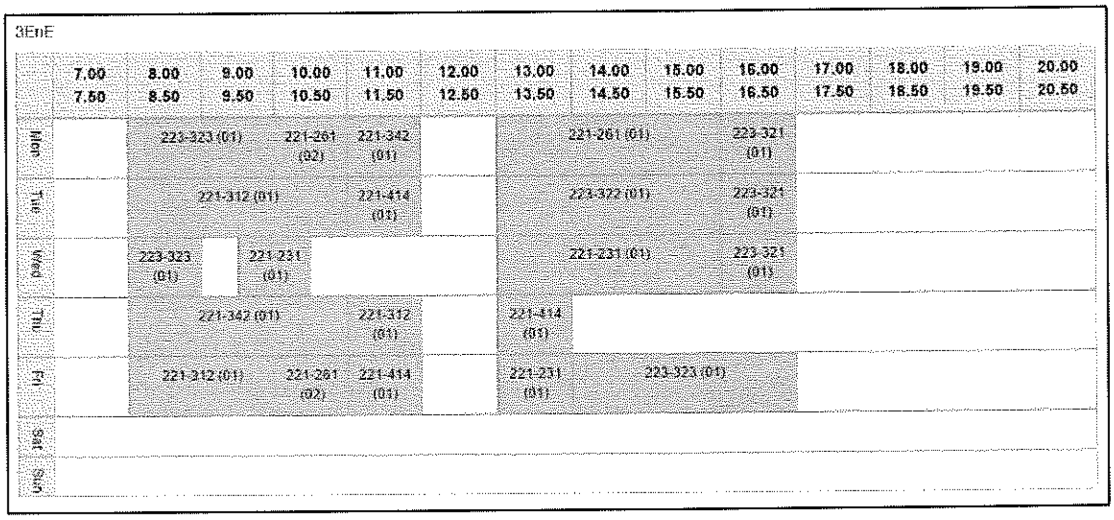


Figure 4-3 3EnE timetable of the second depth DDS results

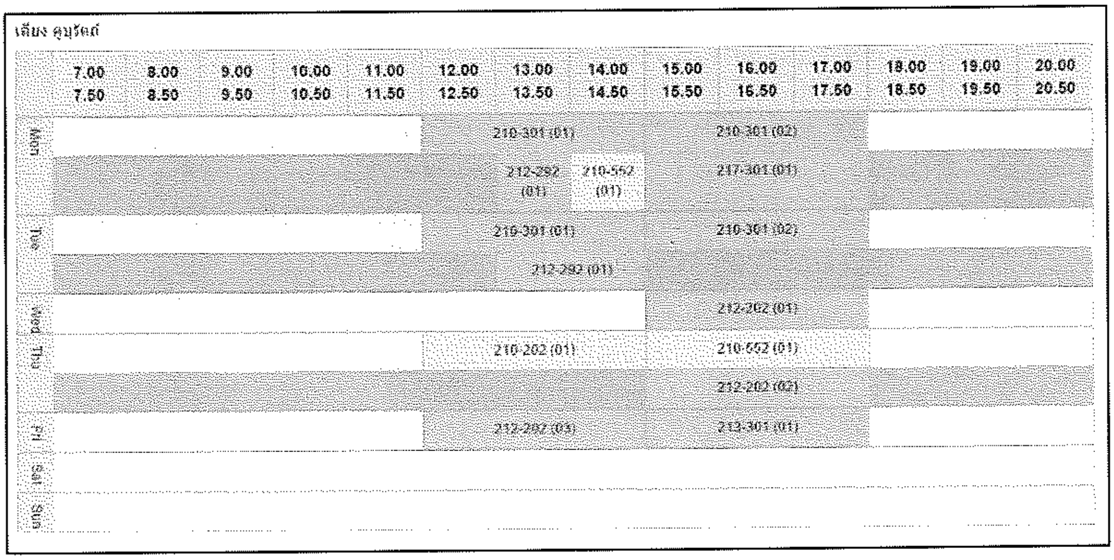


Figure 4-4 Asst.Prof. Leang Khooburat timetable of the UA results

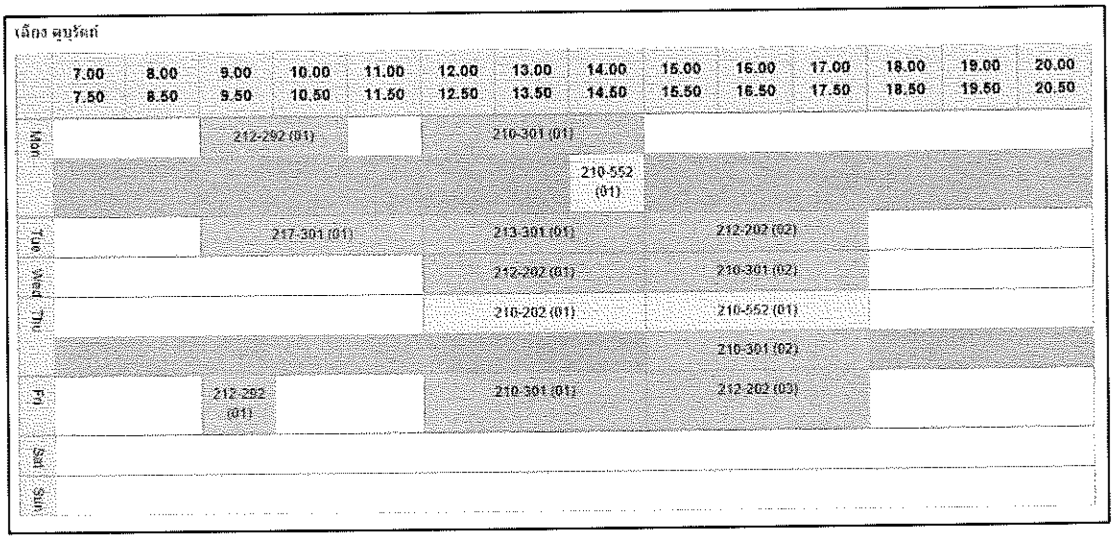


Figure 4-5 Asst.Prof. Leang Khooburat timetable of the first depth DDS results

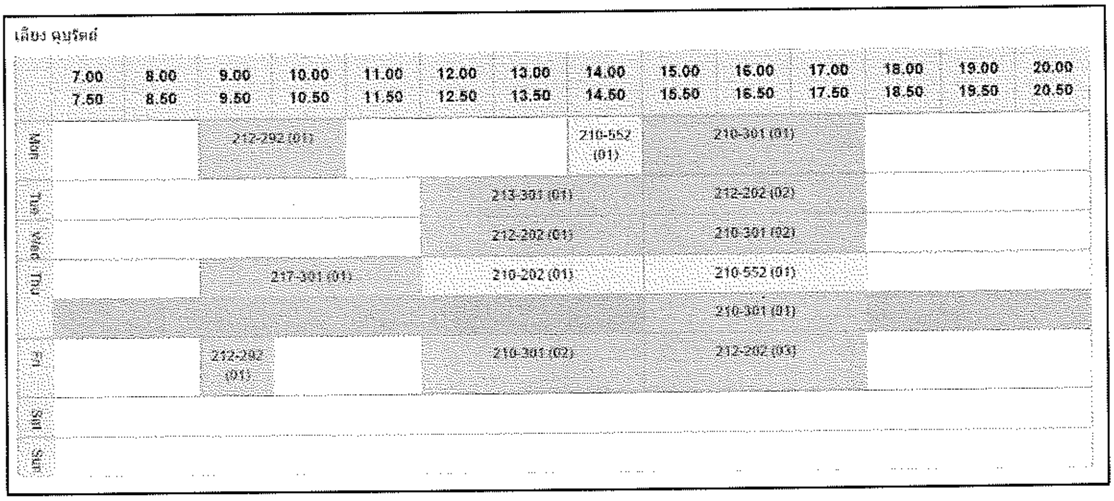


Figure 4-6 Asst.Prof. Leang Khooburat timetable of the second depth DDS results

4.2 University original timetabling analysis

The university original timetabling (UA) are analyzed in Sections 4.2.1 to 4.2.5. Each of which focuses on different group of students. The constraints which are used in the evaluation of the system are also used for evaluating the UA results. The results show that the UA produces a lot of hard constraint violations.

4.2.1 The first year student timetable

The freshman timetables are an interesting case because they contain classes from many faculties such as Faculty of Engineering, Faculty of Science and Faculty of Liberal Art. Table 4-3 shows the constraint violations on the freshman timetables. The most violated hard constraint of freshman timetables is the teacher time conflict constraint which has been violated 27 times. Among the freshman timetables, student group A to I has the highest number of hard constraint violations. The most violated soft constraint of the freshman timetables is the limit-free-timeslot constraint between classes which has been violated 23 times. The constraint of assigning the same section classes on the same timeslot has been violated 20 times. The student group A to I timetables have the highest soft constraint violations among the freshman timetables. The total number of hard constraint violations of the freshman timetables is 27. The total number of soft constraint violations is 66.

Table 4-3 Freshman constraint violations observed from UA

	Hard constraint violations					Soft constraint violations										
	H1	H2	H3	H4	Total	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
Eng A - I	0	0	15	0	15	0	16	14	0	18	0	0	0	3	0	51
Eng J - R	0	0	12	0	12	4	7	2	0	2	0	0	0	0	0	15
Total	0	0	27	0	27	4	23	16	0	20	0	0	0	3	0	66

4.2.2 The second year student timetable

The sophomore timetables contain more laboratory classes in some majors, while the classes from other faculties are reduced. Table 4-4 shows the details of the constraint violations on the sophomore timetables. The most violated hard constraint is as same as that of the freshman timetables, which is the teacher time conflict constraint, 100. The highest number of constraint violations is observed from the computer engineering timetables. The highest number of soft constraint violations is the limit-free-timeslot between classes which has been violated 77 times. The constraint of unassigned same section classes to the adjacent day and the constraint of assigning the same section classes on the same timeslot both have been violated 60 times. The total number of hard constraint violations of the sophomore timetables is 136 and the total number of soft constraint violations is 210.

Table 4-4 Sophomore constraint violations observed from UA

	Hard constraint violations									Soft constraint violations							
	H1	H2	H3	H4	H6	H7	H8	H9	Total	S1	S2	S3	S5	S8	S9	S10	Total
2ME	1	0	20	0	0	0	5	0	26	4	13	5	19	2	0	0	43
2CoE	3	0	29	4	1	3	5	0	45	0	22	15	10	0	1	2	50
2EE	1	2	14	1	0	0	0	1	19	2	7	8	7	0	0	0	24
2IE	0	0	21	2	0	0	5	0	28	0	17	9	15	2	0	0	43
2MnE	0	0	12	0	0	0	0	0	12	0	5	3	4	0	0	0	12
2CE	0	0	0	0	0	1	0	0	1	0	7	11	2	0	0	0	20
2ChE	0	1	4	0	0	0	0	0	5	0	6	9	3	0	0	0	18
Total	5	3	100	7	1	4	15	1	136	6	77	60	60	4	1	2	210

4.2.3 The third year student timetable

Most junior timetables contain laboratory classes, while some elective classes are also presented. Table 4-5 shows the details of the constraint violations on the junior timetables. The most violated hard constraint is the teacher time conflict constraint which has been violated 75 times. The highest hard constraint violations are observed from the mining engineering timetables. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot which has been violated 130 times. The highest soft constraint violations is 63, observed from the mechanical engineering timetables. The next highest number of soft constraint violations is industrial, civil and computer engineering timetables with 62, 58, 58 respectively. The total number of hard constraint violations of the junior timetables is 123 and the total number of soft constraint violations is 330. The mechanical engineering timetables result in the largest number of constraint violations among junior timetables.

Table 4-5 Junior constraint violations observed from UA

	Hard constraint violations							Soft constraint violations							
	H1	H2	H3	H4	H8	H11	Total	S1	S2	S3	S4	S5	S8	S9	Total
3ME	1	1	15	0	9	0	26	0	22	18	0	23	0	0	63
3CoE	0	0	3	2	3	1	9	0	9	28	1	19	0	1	58
3EE	0	1	20	2	0	0	23	0	12	12	0	17	2	0	43
3IE	2	2	11	2	2	0	19	6	17	15	0	24	0	0	62
3MnE	1	0	22	0	1	0	24	0	5	8	0	11	0	2	26
3CE	4	4	2	8	1	0	19	11	1	21	0	24	0	1	58
3ChE	0	1	2	0	0	0	3	0	0	8	0	12	0	0	20
Total	8	9	75	14	16	1	123	17	66	110	1	130	2	4	330

4.2.4 The fourth year student timetable

Senior timetables are similar to those of junior timetables, with a special character. That is, some majors have branches such as computer engineering and electrical engineering. Table 4-6 shows the details of the constraint violations on the senior timetables. The most violated hard constraint is the teacher time conflict constraint, 48. The highest number of hard constraint violations is observed from the electrical engineering timetable, 21. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot, 81. The highest number of soft constraint violations is observed from the civil engineering timetable, 54. The total number of hard constraint violations of senior timetables is 113 and the total number of soft constraint violations is 215.

Table 4-6 Senior constraint violations observed from UA

	Hard constraint violations									Soft constraint violations						
	H1	H2	H3	H4	H5	H8	H9	H10	Total	S1	S2	S3	S5	S8	S9	Total
4ME	1	0	11	5	0	9	0	0	26	6	5	5	16	0	2	34
4CoE	2	3	8	4	0	2	0	0	19	0	23	1	14	0	0	38
4EE	0	0	12	3	0	0	2	3	20	0	16	3	9	3	0	31
4IE	0	0	3	3	0	0	0	0	6	0	6	8	8	0	0	22
4MnE	4	1	10	1	1	0	1	0	18	0	7	6	10	0	2	25
4CE	3	5	2	8	0	1	0	0	19	10	3	21	20	0	0	54
4ChE	1	0	2	2	0	0	0	0	5	0	4	3	4	0	0	11
Total	11	9	48	26	1	12	3	3	113	16	64	47	81	3	4	215

4.2.5 Summary

Table 4-7 shows the total number of hard constraint violations produced by the UA, categorized by groups of students. The total number of hard constraint violations of the UA is 399. The highest violation constraint is the teacher time conflict, 250. Meanwhile, the constraint of assigning the class on regular business hours and the constraint of not to assign the class at noon have been violated 47 and 43 times, respectively.

Table 4-8 shows the total number of soft constraint violations produced by the UA, categorized by groups of students. The total number of soft constraint violations is 821. The constraint of assigning the same section classes on the same timeslot has the highest violation numbers among all. However, the violation of the limit-free-timeslot constraint between classes and the constraint of limit maximum total hour per day follow closely with 233 and 230, respectively.

Table 4-7 The total number of hard constraint violations observed from UA

Student group	Hard constraint violations											
	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	Total
1st year	0	0	27	0	0	0	0	0	0	0	0	27
2nd year	5	3	100	7	0	1	4	15	1	0	0	136
3rd year	8	9	75	14	0	0	0	16	0	0	1	123
4th year	11	9	48	26	1	0	0	12	3	3	0	113
Total	24	21	250	47	1	1	4	43	4	3	1	399

Table 4-8 The total number of soft constraint violations observed from UA

Student group	Soft constraint violations										
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
1st year	4	23	16	0	20	0	0	0	3	0	66
2nd year	6	77	60	0	60	0	0	4	1	2	210
3rd year	17	66	110	1	130	0	0	2	4	0	330
4th year	16	64	47	0	81	0	0	3	4	0	215
Total	43	230	233	1	291	0	0	9	12	2	821

4.3 The results of the first depth of the proposed system

The prototype of the proposed system is developed and evaluated. The ordering heuristic of the proposed system is class-size, teacher-workload and study-hours. To show the potential performance of the proposed system, the results from searching only a discrepancy path (one-depth in the search tree) are shown in this section.

4.3.1 The first year student timetable

Table 4-9 shows the details of constraint violations on freshman timetables. The most violated hard constraint of freshman timetables is the teacher time conflict constraint which has been violated 10 times. The student group A to I timetables violated the hard constraints the most among the freshman timetables. The most violated soft constraint of the freshman timetables is the limit-free-timeslot constraint between classes which has been violated 19 times. The student group A to I timetables violate the soft constraints the most among the freshman timetables. The total number of hard constraint violations is 10 and the total number of soft constraint violations is 46.

Table 4-9 Freshman constraint violations produced by 1st depth DDS

Student group	Hard constraint violations					Soft constraint violations										
	H1	H2	H3	H4	Total	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
Eng A - I	0	0	7	0	7	4	12	0	0	14	0	0	0	0	0	30
Eng J - R	0	0	3	0	3	4	7	0	0	4	0	0	0	0	1	16
Total	0	0	10	0	10	8	19	0	0	18	0	0	0	0	1	46

4.3.2 The second year student timetable

Table 4-10 shows the details of constraint violations on sophomore timetables. The most violated hard constraint of sophomore timetables is the teacher time conflict constraint which has been violated 31 times. However, DDS removes other hard constraint violations. The highest number of hard constraint violations is observed from the computer engineering timetables. The most violated soft constraint is the constraint of not allocating same section classes to be assigned on the adjacent day which has been violated 72 times. Similarly, the computer engineering timetables provide the highest number of soft constraint violations. The total number of hard constraint violations of sophomore timetables is 31 and the total number of soft constraint violations is 184.

Table 4-10 Sophomore constraint violations produced by 1st depth DDS

	Hard constraint violations									Soft constraint violations							
	H1	H2	H3	H4	H6	H7	H8	H9	Total	S1	S2	S3	S5	S8	S9	S10	Total
2ME	0	0	4	0	0	0	0	0	4	0	3	18	17	0	2	0	40
2CoE	0	0	13	0	0	0	0	0	13	2	21	8	12	0	1	2	46
2EE	0	0	3	0	0	0	0	0	3	0	9	1	3	2	0	0	15
2IE	0	0	4	0	0	0	0	0	4	0	1	21	17	0	0	0	39
2MnE	0	0	5	0	0	0	0	0	5	3	1	12	10	0	3	0	29
2CE	0	0	2	0	0	0	0	0	2	0	0	4	1	0	0	0	5
2ChE	0	0	0	0	0	0	0	0	0	0	2	8	0	0	0	0	10
Total	0	0	31	0	0	0	0	0	31	5	37	72	60	2	6	2	184

4.3.3 The third year student timetable

Table 4-11 shows the details of constraint violations on junior timetables. The most violated hard constraint is the teacher time conflict constraint which has been violated 21 times. The electrical engineering timetables produce the highest number of hard constraint violations. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot which has been violated 81 times. The industrial engineering and civil engineering timetables violate the soft constraint the most, 38. The total number of hard constraint violations of junior timetables is 23 and The total number of soft constraint violations is 169. The industrial engineering timetables produce the highest number of constraint violations among all junior timetables.

Table 4-11 Junior constraint violations produced by 1st depth DDS

	Hard constraint violations							Soft constraint violations								
	H1	H2	H3	H4	H8	H11	Total	S1	S2	S3	S4	S5	S8	S9	Total
3ME	0	0	3	0	0	0	3	0	0	3	0	13	0	1	17
3CoE	0	2	0	0	0	0	2	0	8	9	0	0	0	0	17
3EE	0	0	10	0	0	0	10	0	0	9	0	11	3	0	23
3IE	0	0	1	0	0	0	1	0	7	9	0	22	0	0	38
3MnE	0	0	5	0	0	0	5	0	5	4	0	7	0	1	17
3CE	0	0	0	0	0	0	0	2	4	13	0	19	0	0	38
3ChE	0	0	2	0	0	0	2	0	0	10	0	9	0	0	19
Total	0	2	21	0	0	0	23	2	24	57	0	81	3	2	169

4.3.4 The fourth year student timetable

Table 4-12 shows the details of constraint violations on senior timetables. The most violated hard constraint is the teacher time conflict constraint, 32. The highest number of hard constraint violations is 16, observed from the electrical engineering timetables. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot which has been violated 61 times. The highest number of soft constraint violations is 42, observed from the civil engineering timetables. The total number of hard constraint violations is 36 and the total number of soft constraint violations is 134.

Table 4-12 Senior constraint violations produced by 1st depth DDS

	Hard constraint violations									Soft constraint violations						
	H1	H2	H3	H4	H5	H8	H9	H10	Total	S1	S2	S3	S5	S8	S9	Total
4ME	0	0	7	0	0	0	0	0	7	0	5	4	16	0	0	25
4CoE	0	0	0	0	0	0	0	0	0	0	7	10	11	0	0	28
4EE	0	0	13	0	0	0	0	3	16	0	3	5	4	3	0	15
4IE	0	0	1	0	0	0	0	0	1	0	2	2	0	0	0	4
4MnE	0	0	8	0	0	0	0	0	8	0	2	4	7	0	1	14
4CE	0	0	0	1	0	0	0	0	1	5	6	12	19	0	0	42
4ChE	0	0	3	0	0	0	0	0	3	0	0	2	4	0	0	6
Total	0	0	32	1	0	0	0	3	36	5	25	39	61	3	1	134

4.3.5 Summary

In comparison with the UA results, the first depth DDS results reduce a lot of constraint violations. Table 4-13 shows the total hard constraint violations of the first depth DDS, grouped by the students.

A lot of hard constraint violations are reduced to 0. However, some specific cases are violated. For example, the 4EE has to study the electrical laboratory class at noon because of the room availability. The most violated hard constraint is the teacher time conflict constraint which has been violated 94 times. Table 4-14 shows the total soft constraint violations of the first depth DDS, grouped by the students. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot, 220. The other hard constraint and soft constraint violations are shown in Table 4-13 and Table 4-14, respectively.

Table 4-13 The total number of hard constraint violations produced by 1st depth DDS

Student group	Hard constraint violations											
	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	Total
1st year	0	0	10	0	0	0	0	0	0	0	0	10
2nd year	0	0	31	0	0	0	0	0	0	0	0	31
3rd year	0	2	21	0	0	0	0	0	0	0	0	23
4th year	0	0	32	1	0	0	0	0	0	3	0	36
Total	0	2	94	1	0	0	0	0	0	3	0	100

Table 4-14 The total number of soft constraint violations produced by 1st depth DDS

Student group	Soft constraint violations										
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
1st year	8	19	0	0	18	0	0	0	0	1	46
2nd year	5	37	72	0	60	0	0	2	6	2	184
3rd year	2	24	57	0	81	0	0	3	2	0	169
4th year	5	25	39	0	61	0	0	3	1	0	134
Total	20	105	168	0	220	0	0	8	9	3	533

4.4 The results of the second depth of the proposed prototype

The results in this section aim to demonstrate the performance improvement of the proposed prototype when the search time is increasing. By allowing the DDS to explore its discrepancy on the second depth, the improvement can be evaluated in terms of time versus performance.

The ordering heuristic is as same as that used in Section 4.3. However, the DDS searches deeper into the paths of the discrepancy in the second depth of the tree.

4.4.1 The first year student timetable

Table 4-15 shows the details of the constraint violations on the freshman timetables. The most violated hard constraint of freshman timetables is the teacher time conflict constraint which has been violated 3 times. The most violated soft constraint of assigning the same section classes on the same timeslot which has been violated 17 times. The student group A to I timetables produce the highest number of the total violations among all freshman timetables. The total number of hard constraint violations is 3 and the total number of soft constraint violations is 35.

Table 4-15 Freshman constraint violations produced by 2nd depth DDS

	Hard constraint violations					Soft constraint violations										
	H1	H2	H3	H4	Total	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
Eng A - I	0	0	0	0	0	1	7	0	0	14	0	0	0	1	0	23
Eng J - R	0	0	3	0	3	2	6	0	0	3	0	0	0	0	1	12
Total	0	0	3	0	3	3	13	0	0	17	0	0	0	1	1	35

4.4.2 The second year student timetable

Table 4-16 shows the details of the constraint violations on the sophomore timetables. The most violated hard constraint of sophomore timetables is the teacher time conflict constraint which has been violated 36 times. The violation is increased from the first depth DDS. The mechanical engineering timetables violate the hard constraints the most. The most violated soft constraint is the constraint of not allocating same section classes on adjacent days which has been violated 58 times. The total number of hard constraint violations is 36 and the total number of soft constraint violations is 148. The industrial engineering timetables produce the highest total number of constraint violations, 51.

Table 4-16 Sophomore constraint violations produced by 2nd depth DDS

	Hard constraint violations									Soft constraint violations							
	H1	H2	H3	H4	H6	H7	H8	H9	Total	S1	S2	S3	S5	S8	S9	S10	Total
2ME	0	0	12	0	0	0	0	0	12	0	2	16	15	0	2	0	35
2CoE	0	0	6	0	0	0	0	0	6	0	0	8	17	0	2	2	29
2EE	0	0	2	0	0	0	0	0	2	0	9	1	3	2	0	0	15
2IE	0	0	10	0	0	0	0	0	10	0	8	17	16	0	0	0	41
2MnE	0	0	4	0	0	0	0	0	4	0	2	12	4	0	3	0	21
2CE	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	2
2ChE	0	0	1	0	0	0	0	0	1	0	1	4	0	0	0	0	5
Total	0	0	36	0	0	0	0	0	36	0	23	58	56	2	7	2	148

4.4.3 The third year student timetable

Table 4-17 shows the details of the constraint violations on the junior timetables. The most violated hard constraint is the teacher time conflict constraint which has been violated 17 times. The electrical engineering timetables produce the highest number of hard constraint violations, 7. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot which has been violated 93 times. The industrial engineering timetables produce the highest number of soft constraint violations, 41. The total number of hard constraint violations is 17 and the total number of soft constraint violations is 179. The industrial engineering timetables produce the highest total constraint violations among all junior timetables.

Table 4-17 Junior constraint violations produced by 2nd depth DDS

	Hard constraint violations							Soft constraint violations							
	H1	H2	H3	H4	H8	H11	Total	S1	S2	S3	S4	S5	S8	S9	Total
3ME	0	0	1	0	0	0	1	0	7	4	0	12	0	1	24
3CoE	0	0	3	0	0	0	3	0	8	7	0	0	0	0	15
3EE	0	0	7	0	0	0	7	0	6	5	0	18	2	0	31
3IE	0	0	1	0	0	0	1	0	4	9	0	28	0	0	41
3MnE	0	0	3	0	0	0	3	0	4	1	0	9	0	2	16
3CE	0	0	0	0	0	0	0	0	9	8	0	20	0	0	37
3ChE	0	0	2	0	0	0	2	0	0	9	0	6	0	0	15
Total	0	0	17	0	0	0	17	0	38	43	0	93	2	3	179

4.4.4 The fourth year student timetable

Table 4-18 shows the details of the constraint violations on the senior timetables. The most violated hard constraint is the teacher time conflict constraint, 31. The electrical engineering timetables produce the highest number of hard constraint violations at 13. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot which has been violated 61 times. The mechanical engineering timetables produce the highest number of total violations, 55. The total number of hard constraint violations is 34 and the total number of soft constraint violations is 144.

Table 4-18 Senior constraint violations produced by 2nd depth DDS

	Hard constraint violations									Soft constraint violations						
	H1	H2	H3	H4	H5	H8	H9	H10	Total	S1	S2	S3	S5	S8	S9	Total
4ME	0	0	7	0	0	3	0	0	10	0	15	8	21	0	1	45
4CoE	0	0	0	0	0	0	0	0	0	0	6	6	8	0	0	20
4EE	0	0	13	0	0	0	0	0	13	0	4	2	7	3	0	16
4IE	0	0	2	0	0	0	0	0	2	0	0	1	0	0	0	1
4MnE	0	0	4	0	0	0	0	0	4	0	7	3	4	0	1	15
4CE	0	0	1	0	0	0	0	0	1	0	12	13	21	0	0	46
4ChE	0	0	4	0	0	0	0	0	4	0	0	1	0	0	0	1
Total	0	0	31	0	0	3	0	0	34	0	44	34	61	3	2	144

4.4.5 Summary

Table 4-19 shows the total number of hard constraint violations produced by the second depth DDS which reduces a few hard constraint violations from that of the first depth DDS results. The hard constraint violations of H2 and H4 are reduced to 0. The most violated hard constraint is the teacher time conflict constraint which has been violated 87 times, which is 7 time less than that of the first depth result. Table 4-20 shows the total number of soft constraint violations produced by the second depth DDS. The most violated soft constraint is the constraint of assigning the same section classes on the same timeslot which has been violated 227 times, which is 7 more time than that of the first depth result.

Table 4-19 The total number of hard constraint violations produced by 2nd depth DDS

Student group	Hard constraint violations											
	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	Total
1st year	0	0	3	0	0	0	0	0	0	0	0	3
2nd year	0	0	36	0	0	0	0	0	0	0	0	36
3rd year	0	0	17	0	0	0	0	0	0	0	0	17
4th year	0	0	31	0	0	0	0	3	0	0	0	34
Total	0	0	87	0	0	0	0	3	0	0	0	90

Table 4-20 The total number of soft constraint violations produced by 2nd depth DDS

Student group	Soft constraint violations										
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	Total
1st year	3	13	0	0	17	0	0	0	1	1	35
2nd year	0	23	58	0	56	0	0	2	7	2	148
3rd year	0	38	43	0	93	0	0	2	3	0	179
4th year	0	44	34	0	61	0	0	3	2	0	144
Total	3	118	135	0	227	0	0	7	13	3	506

4.5 Hard/soft constraint relation

This section evaluates the effect of three objective models of the constraints. The first model, simple mixed objective (SMO), assigns an equal priority to all soft constraints. The second model, mixed objective 1 (MO1), assigns a higher priority to time constraints (TC). The third model, mixed objective 2 (MO2), assigns a higher priority to time-spread constraints (TSC). Table 3-1 and Table 3-2 show the details of each constraint. Time constraints include S8, S9 and S10, while time-spread constraints include S1, S2, S3, S4 and S5. Hard constraints used are H1, H3, H4, H5, H6, H7, H8, H9 and H10.

Table 4-21 shows the soft constraint violation results of all three objective models while Table 4-22 shows the hard and soft constraint violation results of all three objective models. At the first look, the hard constraint violations of all three results from the proposed prototype are reduced from that of the UA results presented in Table 4-7.

Table 4-21 Soft constraint violations grouped by models

	TSC					TC				
Model	S1	S2	S3	S5	Total	S8	S9	S10	Total	Overall
SMO	4	61	100	182	347	8	10	3	21	368
MO1	3	89	98	176	366	6	6	3	15	381
MO2	2	57	106	161	326	8	12	9	29	355

Table 4-22 Hard/soft constraint violations categorized by student groups

	SMO			MO1			MO2		
Student group	HC	TSC	TC	HC	TSC	TC	HC	TSC	TC
1st Year	0	26	1	0	26	1	0	15	8
2nd Year	4	100	10	4	122	5	4	89	11
3rdYear	10	138	5	12	147	6	10	147	5
4th Year	34	83	5	36	71	3	40	75	5
Total	48	347	21	52	366	15	54	326	29

The equal priority soft constraint SMO cannot dominate the other model on any soft constraint violation. However, SMO produces the lowest number of hard constraint violations (HC) as shown in Table 4-22. MO1 and MO2 show the effect of mixed objective models. MO1 which assigns a high priority to time constraints, produces the lowest number of time constraint violations. MO2 which assigns a high priority to time-spread constraints produces the lowest number of time-spread constraint violations.

According to the results of MO1, the total number of TC violations is the lowest among all results because MO1 gives a high priority to the TC type, while the results of MO2 show the lowest total number of TSC violations among all results. Thus, this results show that each objective model can guide the search engine to a suitable solution according to the specified objective.

The freshman timetables are better when MO2 is applied. Since, the freshman timetables have limited available slots, the restriction on time-spread constraints is more effective. The sophomore timetables are considered early in the search process. Therefore, there are a lot of available spots. Unless the sophomore timetables, the junior and senior timetables are considered later in the search process. Thus, there are also limited spots. However, MO1 shows a few number of hard constraint violations in comparison with that produced by MO2.

4.6 Ordering heuristic and dataset

There are two experiments in this section. The first set of experiments is focusing on evaluating the ordering heuristic on the original dataset. The results of the first experiment are shown in Section 4.6.1. The second set of experiments is focusing on evaluating the effects of ordering heuristics on the modified datasets in order to provide evidents to draw a conclusion. The results of the second experiments are shown in Section 4.6.2

4.6.1 Original datasets

This section focuses on the effect of ordering heuristics on different characteristics of the datasets. There are six ordering heuristics created by a combination of class size, teacher workload and study-hour as shown in Table 4-23.

Table 4-23 Ordering heuristic definition

Heuristic index	Ordering heuristic
OH1	Class size > Teacher > Study-hour
OH2	Class size > Study-hour > Teacher
OH3	Teacher > Class size > Study-hour
OH4	Teacher > Study-hour > Class size
OH5	Study-hour > Class size > Teacher
OH6	Study-hour > Teacher > Class size

These ordering heuristics are applied in order to create a search tree. To minimize the effect of the objective model, the SMO is used and all hard constraint violations are combined as a single value (HC) while all soft constraint violations are also combined as a single value (SC). Table 4-24 shows the results on the freshman datasets. The ordering heuristic does not affect the Eng A - I students while it increases the soft constraint violations in Eng J - R students by 2. Thus, the class size must be given a higher priority than that of the study-hour in case of the freshman timetables.

Table 4-24 Freshman timetables constraint violations of different ordering heuristics

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
Eng A - I	0	16	0	16	0	16	0	16	0	16	0	16
Eng J - R	0	11	0	11	0	11	0	13	0	13	0	13

Table 4-25 shows the number of constraint violations of sophomore timetables of different ordering heuristics. The sophomore timetables also show similar number of constraint violations except the 2ME and 2IE dataset. 2ME results show a better performance when the class size does not have a high priority while 2IE shows a better performance when the teacher workload has a high priority.

Table 4-25 Sophomore timetables constraint violations of different ordering heuristics

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
2ME	0	40	0	39	0	28	0	31	0	30	0	30
2CoE	4	29	4	28	4	28	4	31	4	25	4	26
2EE	0	6	0	6	0	7	0	5	0	5	0	5
2IE	0	17	0	19	0	15	0	12	0	21	0	21
2MnE	0	12	0	14	0	13	0	10	0	10	0	10
2CE	0	2	0	2	0	2	0	2	0	2	0	2
2ChE	0	7	0	7	0	6	0	6	0	6	0	3

Table 4-26 shows the number of constraint violations of the sophomore timetable of the different ordering heuristics. The effects of ordering heuristic are clearly visible on junior timetables. 3ME and 3IE datasets have a better performance when the study-hour gets a high priority. 3EE dataset achieves a low number of hard constraint violations when the class size gets a high priority. 3MnE dataset achieves a better performance when the study-hour does not get a high priority. Significantly, the OH2, OH3 and OH4 do not produce a complete result on 3CE dataset, which seems to react well to the high priority on study-hour. 3ChE dataset achieves a better performance when study-hour gets a higher priority than the class-size.

Table 4-26 Junior timetables constraint violations of different ordering heuristics

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
3ME	0	17	0	15	0	11	0	13	0	7	0	9
3CoE	2	8	2	9	2	12	2	12	2	9	2	8
3EE	7	29	7	29	9	27	11	21	13	23	14	21
3IE	0	41	1	31	0	41	0	37	0	29	0	28
3MnE	1	9	1	11	1	9	4	10	1	15	1	13
3CE	0	30	-	-	-	-	-	-	0	26	0	26
3ChE	0	18	0	18	0	18	0	12	0	14	0	15

Table 4-27 shows the results of constraint violations producing under different ordering heuristic on senior timetables. 4CoE and 4IE datasets have a better performance when the study-hour gets a high priority. 4ME and 4EE datasets provide a similar number of violations on all ordering. 4MnE dataset achieves a better performance from OH4. 4CoE dataset seems to provide a better performance when the class size does not get a high priority. 4CE dataset achieves a better performance when a high priority is given to the teacher workload.

Table 4-27 Senior timetables constraint violations of different ordering heuristics

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
4ME	8	30	6	26	9	31	7	21	7	28	4	25
4CoE	0	18	0	20	0	14	0	13	0	10	0	6
4EE	16	11	16	6	21	10	19	9	16	11	16	10
4IE	1	3	2	2	1	4	1	4	0	4	0	4
4MnE	8	14	3	18	8	16	1	12	7	9	11	12
4CE	1	15	1	15	0	10	0	10	1	11	2	21
4ChE	4	5	5	4	2	9	7	7	5	2	5	2

DDS with OH1, OH5 and OH6 can produce a complete result for all datasets. The performance comparison between freshman, sophomore, junior and senior timetables is shown in Table 4-28. The number of freshman timetable constraint violations is similar on all ordering heuristics. OH5 and OH6 produce better performance on sophomore datasets. Junior timetables have lowest hard constraint violations in OH1. However, the soft constraint is also the most violated in OH1. The senior timetables is better on OH5 which considering the study-hour before the class size and less considering the teacher workload.

Table 4-28 Comparison total constraint violations of different ordering heuristics

Student Group	Number of violations					
	OH1		OH5		OH6	
	HC	SC	HC	SC	HC	SC
1st year	0	27	0	29	0	29
2nd year	4	113	4	99	4	97
3rd year	10	152	16	123	17	120
4th year	38	96	36	75	38	80
Total	52	388	56	326	59	326

4.6.2 Modified dataset

To evaluate the effect of ordering heuristics on different dataset characteristics, the UAs are modified into different characteristic datasets as testing dataset (TD). There are 4 modified datasets including TD1, TD2, TD3 and TD4. The original dataset is modified by splitting all two-hour, three-hour classes to one-hour classes, in TD1. Thus, TD1 dataset contains a lot of short-class types. TD2, TD3 and TD4 datasets are modified in a similar fashion. However, TD2 splits classes to 1.5-hour classes. TD3 splits and combines all classes to 2-hour classes and TD4 combines all classes to 3-hour classes. Table 4-30 shows the characteristics of all datasets. The experiments are conducted on 5 datasets (UA, TD1, TD2, TD3 and TD4) and 6 different ordering heuristics (OH1, OH2, OH3, OH4, OH5 and OH6). Thus, there are 30 sets of results.

Table 4-29 Description of datasets

Dataset	Description
UA	University original timetabling
TD1	1-hour lecture type classes
TD2	1.5-hour lecture type classes
TD3	2-hour lecture type classes
TD4	3-hour lecture type classes

Table 4-30 Characteristics of datasets grouped by subject types

Dataset	Number of classes																	
	Lab						Main						Elective					
	1	1.5	2	2.5	3	4	1	1.5	2	2.5	3	4	1	1.5	2	2.5	3	4
UA	33	2	36	1	71	1	284	43	106	6	30	4	37	8	14	0	10	0
TD1	73	0	18	0	72	1	597	11	17	6	9	2	106	0	1	0	1	0
TD2	8	6	46	0	72	1	103	307	42	6	9	2	7	62	4	0	1	0
TD3	5	0	52	0	72	1	187	11	222	6	9	2	20	0	44	0	1	0
TD4	5	0	52	0	72	1	85	11	120	6	111	2	1	0	25	0	20	0

Table 4-31 shows the results of all datasets of freshman timetables. A complete result can be produced for all datasets. Eng A - I students achieve good results under TD2 and TD3, While Eng J - R students achieve good results under OH4, OH5 and OH6. However, the score of all datasets is not greatly different.

Table 4-32 shows the results of all sophomore datasets. TD4 shows good performances on all datasets except 2ChE. The difference between the best and the average score is large in the datasets with large total hour characteristics, including 2ME, 2CoE, 2IE and 2MnE.

Table 4-33 shows the results of all junior datasets. Three datasets cannot be completed in this case including 3ME, 3IE and 3CE. These datasets have large total hour characteristics. Especially, 3CE with 6 characteristics has only 10 complete results out of 30 sets. TD3 and TD4 produce good performance except on 3EE dataset. 3ME, 3IE and 3MnE provide a large performance difference between the best and the average scores.

Table 4-34 shows the results of all senior datasets. Five datasets cannot be completed in this case including 4ME, 4IE, 4MnE, 4CE and 4ChE. These datasets are considered difficult. 4ME dataset contains many characteristics. 4MnE dataset contains many multiple teachers. 4IE dataset is classified as a high workload teacher. 4CE dataset has the highest total hours among others. On the other hand, 4CoE and 4EE datasets contain the branch curriculums, making them less difficult than the others.

Table 4-35 shows the results of different ordering heuristics on TD1. In comparison with other datasets, TD1 produces the large number of violations because TD1 contains a lot of classes. Five groups of students cannot be completed under TD1, including 3IE, 3CE, 4IE, 4MnE and 4CE. Table 4-36 shows the results of TD2. Table 4-37 shows the results of TD3. Table 4-38 shows the results of TD4. In comparison, TD4 contains less number of classes with the biggest class size mostly. TD4 produces less number of violations due to the less number of classes. TD1 causes issues with soft constraints especially not-allow class to be allocated on the same day.

TD3 with OH5 produces the best total score in comparison with the other datasets. TD3 with OH6 also produces similar total score, with a slightly different number of soft constraint violations.

OH5 and OH6, emphasizing on the study-hour constraint provide a better score on most datasets. OH3 and OH4, emphasizing on the teacher workload constraint provide a better score on multiple-teacher datasets and allowed-to-have-teacher-time-conflict datasets. OH1 and OH2, emphasizing on the class size constraint, provide a better score on large-class-size datasets such as TD4.

Some dataset characteristics have no effect on the selected ordering heuristic but the difficulty in complete such datasets is increased. Such characteristics include room-size, multiple-room and high workload teacher. Especially, the high workload teacher characteristic causes the datasets to be too difficult to complete (3CE, 3IE, 4CE, 4IE and 4MnE).

Table 4-31 Results of freshman modified datasets

	Has complete results	No complete result		Best score results			Average score	
		Number	Details	HC	SC	Details	HC	SC
Eng A - I	30	0	-	0	4	TD2:OH1; TD2:OH2; TD2:OH3; TD2:OH4; TD2:OH5; TD2:OH6; TD3:OH1; TD3:OH2; TD3:OH3; TD3:OH4; TD3:OH5; TD3:OH6	0	9.3
Eng J - R	30	0	-	0	10	TD2:OH4; TD2:OH5; TD2:OH6; TD3:OH4; TD3:OH5; TD3:OH6; TD4:OH4; TD4:OH5; TD4:OH6	0	13.2

Table 4-32 Results of sophomore modified datasets

	Has complete results	No complete result		Best score results			Average score	
		Number	Details	HC	SC	Details	HC	SC
2ME	30	0	-	0	4	TD4:OH4	0.1	22.53
2CoE	30	0	-	0	13	TD4:OH5	0.87	30.07
2EE	30	0	-	0	2	TD4:OH1; TD4:OH2; TD4:OH3; TD4:OH4; TD4:OH5; TD4:OH6	0	6.73
2IE	30	0	-	0	4	TD4:OH1	0	16.5
2MnE	30	0	-	0	3	TD4:OH1	0	12.6
2CE	30	0	-	0	0	TD4:OH1; TD4:OH3; TD4:OH4	0	4.33
2ChE	30	0	-	0	1	TD3:OH4	0.06	5.4

Table 4-33 Results of junior modified datasets

	Has complete results	No complete result		Best score results			Average score	
		Number	Details	HC	SC	Details	HC	SC
3ME	29	1	TD1:OH5	0	6	TD3:OH4	0.2	14.45
3CoE	30	0	-	0	2	TD4:OH6	1.13	6.23
3EE	30	0	-	7	29	UA:OH1; UA:OH2; UA:OH3	12.23	28.2
3IE	24	6	TD1:OH1; TD1:OH2; TD1:OH3; TD1:OH4; TD1:OH5; TD1:OH6	0	8	TD4:OH5	0.04	25.67
3MnE	30	0	-	0	3	TD3:OH5	1	11.43
3CE	10	20	UA:OH2; UA:OH3; UA:OH4; TD1:OH1; TD1:OH2; TD1:OH3; TD1:OH4; TD1:OH5; TD1:OH6; TD2:OH1; TD2:OH2; TD2:OH3; TD2:OH4; TD2:OH5; TD2:OH6; TD3:OH3; TD3:OH4; TD4:OH1; TD4:OH2; TD4:OH3; TD4:OH5	0	10	TD3:OH2	0.6	17.9
3ChE	30	0	-	0	1	TD3:OH1; TD3:OH2; TD3:OH3; TD3:OH4; TD3:OH5; TD4:OH2; TD4:OH3; TD4:OH4; TD4:OH6	0	7

Table 4-34 Results of senior modified datasets

	Has complete results	No complete result		Best score results			Average score	
		Number	Details	HC	SC	Details	HC	SC
4ME	22	8	TD2:OH3; TD2:OH4; TD2:OH5; TD3:OH2; TD4:OH3; TD4:OH4; TD4:OH5; TD4:OH6	4	25	UA:OH6	9.45	40.5
4CoE	30	0	-	0	0	TD3:OH1; TD3:OH2; TD3:OH3; TD3:OH5; TD3:OH6; TD4:OH2; TD4:OH5; TD4:OH6	0	6.43
4EE	30	0	-	9	3	TD4:OH5	14.63	12.33
4IE	21	9	TD1:OH1; TD1:OH2; TD1:OH3; TD1:OH4; TD1:OH5; TD1:OH6; TD4:OH1; TD4:OH3; TD4:OH6	0	2	TD4:OH4	0.33	10.48
4MnE	23	7	TD1:OH1; TD1:OH2; TD1:OH3; TD1:OH4; TD1:OH5; TD1:OH6; TD2:OH2	1	12	UA:OH4	9.04	11.52
4CE	17	13	TD1:OH1; TD1:OH2; TD1:OH3; TD1:OH4; TD1:OH5; TD1:OH6; TD2:OH3; TD2:OH4; TD2:OH5; TD4:OH2; TD4:OH3; TD4:OH5; TD4:OH6	0	7	TD3:OH4	1.47	9.18
4ChE	27	3	TD2:OH6; TD4:OH3; TD4:OH4	0	24	TD1:OH4	3.04	11.52

Table 4-35 Constraint violations observed from TD1 datasets

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
Eng A - I	0	16	0	16	0	16	0	15	0	15	0	15
Eng J - R	0	19	0	19	0	19	0	22	0	22	0	22
2ME	1	38	1	33	1	38	0	48	0	53	0	53
2CoE	0	65	0	61	0	71	0	71	0	61	0	66
2EE	0	19	0	19	0	19	0	19	0	19	0	19
2IE	0	25	0	28	0	25	0	34	0	43	0	43
2MnE	0	24	0	15	0	24	0	14	0	16	0	16
2CE	0	9	0	12	0	9	0	9	0	9	0	9
2ChE	0	12	0	16	0	12	0	12	0	14	0	14
3ME	1	43	0	24	1	43	0	29	-	-	1	28
3CoE	1	11	1	14	1	17	1	6	1	19	1	6
3EE	12	57	12	50	12	53	11	53	21	65	16	38
3IE	-	-	-	-	-	-	-	-	-	-	-	-
3MnE	1	22	2	28	1	27	0	30	1	23	1	22
3CE	-	-	-	-	-	-	-	-	-	-	-	-
3ChE	0	10	0	15	0	13	0	13	0	10	0	10
4ME	8	68	9	108	11	86	18	78	9	85	8	68
4CoE	0	12	0	13	0	17	0	14	0	9	0	9
4EE	15	26	12	37	18	28	12	25	12	38	18	30
4IE	-	-	-	-	-	-	-	-	-	-	-	-
4MnE	-	-	-	-	-	-	-	-	-	-	-	-
4CE	-	-	-	-	-	-	-	-	-	-	-	-
4ChE	1	25	1	36	0	26	0	24	1	23	1	22

Table 4-37 Constraint violations observed from TD3 datasets

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
Eng A - I	0	4	0	4	0	4	0	4	0	4	0	4
Eng J - R	0	13	0	13	0	13	0	10	0	10	0	10
2ME	0	14	0	14	0	14	0	12	0	11	0	11
2CoE	0	18	0	18	0	16	0	16	0	16	1	19
2EE	0	3	0	3	0	3	0	3	0	3	0	3
2IE	0	12	0	12	0	12	0	11	0	11	0	11
2MnE	0	15	0	17	0	15	0	14	0	14	0	14
2CE	0	1	0	1	0	1	0	1	0	1	0	1
2ChE	0	2	0	3	0	2	0	1	0	3	0	4
3ME	0	10	0	11	0	9	0	6	0	9	0	10
3CoE	1	3	1	3	1	2	1	2	1	2	0	3
3EE	11	16	10	18	12	20	12	17	11	21	10	21
3IE	0	20	0	22	0	20	0	31	0	19	0	17
3MnE	1	8	0	11	1	6	0	7	0	3	0	7
3CE	0	22	0	10	-	-	-	-	0	13	0	15
3ChE	0	1	0	1	0	1	0	1	0	1	0	2
4ME	16	17	-	-	11	24	6	23	6	19	9	17
4CoE	0	0	0	0	0	0	0	1	0	0	0	0
4EE	15	5	15	12	9	8	15	8	15	7	15	2
4IE	0	9	0	5	0	10	0	10	0	7	0	9
4MnE	9	10	11	4	8	11	9	11	9	11	8	14
4CE	1	12	1	3	2	9	0	7	1	4	1	5
4ChE	1	8	1	3	1	8	1	5	2	9	1	5

Table 4-38 Constraint violations observed from TD4 datasets

Dataset	Number of violations											
	OH1		OH2		OH3		OH4		OH5		OH6	
	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC	HC	SC
Eng A - I	0	6	0	6	0	6	0	6	0	6	0	6
Eng J - R	0	13	0	13	0	13	0	10	0	10	0	10
2ME	0	9	0	12	0	9	0	4	0	5	0	5
2CoE	0	16	0	16	0	15	0	14	0	13	1	16
2EE	0	2	0	2	0	2	0	2	0	2	0	2
2IE	0	4	0	6	0	6	0	6	0	6	0	8
2MnE	0	3	0	4	0	11	0	11	0	4	0	6
2CE	0	0	0	1	0	0	0	0	0	1	0	1
2ChE	0	2	0	3	1	1	1	1	0	3	0	3
3ME	0	7	0	8	0	9	0	7	0	8	0	7
3CoE	1	1	1	1	1	1	1	1	1	1	0	2
3EE	10	25	10	21	19	19	13	23	9	21	8	15
3IE	0	20	0	18	0	24	0	14	0	8	0	13
3MnE	1	3	2	5	2	3	2	4	3	5	2	4
3CE	-	-	-	-	-	-	2	9	-	-	2	7
3ChE	0	3	0	1	0	1	0	1	0	3	0	1
4ME	12	14	9	16	-	-	-	-	-	-	-	-
4CoE	0	1	0	0	0	2	0	2	0	0	0	0
4EE	15	4	12	4	12	4	14	3	9	3	14	4
4IE	-	-	1	8	-	-	0	2	1	9	-	-
4MnE	8	10	13	10	11	5	6	5	6	12	12	8
4CE	2	6	-	-	-	-	2	1	-	-	-	-
4ChE	3	7	2	8	-	-	-	-	7	3	7	6

4.7 Performance of the prototype

The proposed system is evaluated on a machine with Intel(R) Core(TM) i5-2400 3.10 GHz Central Processing Unit (CPU) and 8 Gigabyte memory. The runtime result is shown in Table 4-39. The table shows the number of paths and the total number of nodes in the search tree of each dataset that are considered by the search process. The average runtime on each node of the first path is 0.616 seconds. The maximum runtime per node of the first path is 0.690 seconds, from the 2MnE dataset. The longest path solution is 51 nodes from the 2CoE dataset. Due to the caching technique, the processing time per node is reduced to 0.462 seconds on the path of depth 1. Thus, the total time to search the discrepancy path in this dataset is 4 minutes, without pruning any path. This is the time it takes to get the first solution in this experiment.

Table 4-39 Runtime of the proposed system

Dataset	Discrepancy level							
	depth = 0 (First Path)				depth = 1			
	Path number	Node	Time (sec)	Time/node (sec)	Path number	Node	Time (sec)	Time/node (sec)
Eng A - I	1	21	13.657	0.650	18	378	229.8	0.608
Eng J - R	1	14	8.448	0.603	13	182	52.114	0.286
2CoE	1	51	31.593	0.619	22	1,122	226.8	0.202
2EE	1	17	11.147	0.656	16	272	133.8	0.492
2ME	1	35	21.387	0.611	9	315	153.5	0.487
2CE	1	17	10.684	0.628	1	17	6.872	0.404
2IE	1	19	11.609	0.611	2	38	21.882	0.576
2ChE	1	8	4.364	0.546	7	56	23.42	0.418
2MnE	1	15	10.355	0.690	8	120	74.6	0.622
3CoE	1	33	20.547	0.623	1	33	18.68	0.566
3EE	1	35	22.164	0.633	4	140	7.148	0.051
3ME	1	31	19.542	0.630	4	124	45.991	0.371

Table 4-39 Runtime of the proposed system (cont.)

Dataset	Discrepancy level							
	depth = 0 (First Path)				depth = 1			
	Path number	Node	Time (sec)	Time/node (sec)	Path number	Node	Time (sec)	Time/node (sec)
3CE	1	37	22.551	0.609	12	444	226.3	0.510
3IE	1	32	19	0.594	8	256	146.8	0.573
3ChE	1	17	10.579	0.622	16	272	134.8	0.496
3MnE	1	23	13.68	0.595	7	161	86.5	0.537
4CoE	1	22	13.621	0.619	1	22	8.15	0.370
4EE	1	29	17.926	0.618	24	696	328.9	0.473
4ME	1	26	16.457	0.633	4	104	65.7	0.632
4CE	1	28	17.101	0.611	5	140	79.3	0.566
4IE	1	17	9.606	0.565	10	170	48.24	0.284
4ChE	1	10	5.904	0.590	9	90	44.742	0.497
4MnE	1	24	14.448	0.602	4	96	57.698	0.601
			Average	0.616			Average	0.462

4.8 Discussion

The original timetables provided by the university shown in Section 2.2 displays a lot of time conflicts on both teachers and students. Some hard constraints are also violated. The total number of hard constraint violations is shown in Table 4-7. As a result, many problems occur when the timetables are put in action.

Section 4.2 and 4.3 present a better solution produced by the first and the second depth DDS. The total number of hard constraint violations is reduced from 399 to 100 and 90, respectively. The total number of soft constraint violations is also reduced from 821 to 533 and 506, respectively. Section 4.3 shows that the overall number of constraint violations of the second depth DDS are lower than that of the first depth DDS. However, the junior and senior student timetables incur a large number of soft constraint violations in comparison with that of the first depth. Although, the results of the total violations of the second depth DDS on some datasets are worse than that of the first depth DDS. The total number of hard constraint violations is reduced from 100 to 90. Thus, the deeper depth DDS will provides a better solution.

Section 4.5 presents the effect of three objective models. Table 4-22 shows that MO1 which emphasized on the time constraints produces a better solution than that of the MO2 which emphasized on the time-spread constraints. Thus, the time constraints have more effects on the proposed system than the time-spread constraints. However, the SMO results presented in Section 4.5 which regards all constraints as equal, show a promising score. Therefore, an equal priority can help in avoiding the effects of constraint types for the datasets that are complex.

Table 4-40 shows the characteristics of each dataset. Section 4.6 present the effect of ordering heuristics on different datasets. The results show that OH1 can complete the schedule for all datasets and OH1 can complete most of the modified datasets, in comparison with other ordering heuristics. OH2 performs well for 4EE and 4IE datasets. While, OH2 cannot finish the schedule for 3CE dataset. Thus, if the class-size constraint is already considered the teacher load constraint is a better next-order constraint to be used than the study-hour constraint. OH3 and OH4 emphasized on the teacher workload constraint clearly achieve good performances on 2ME and 3ChE datasets. Both heuristics also provide a good performance on the other datasets. However, the performance differences among ordering heuristics are not significant. Moreover, OH3 and OH4 fail to produce a complete result for 3CE dataset. OH5 and OH6 produce similar performance. 2CoE and 2ME datasets achieve a good performance under OH5 and OH6, because both datasets contain a lot of study hours.

The performance of all 6 ordering heuristics on modified datasets show that the workloads with a large number of small class-size classes in the set might run into an issue with time-spread soft constraints such as the limitation on the same class to be scheduled on the adjacent day. A lot of classes will increase the difficulty on the later classes. As a result, no ordering heuristic provides a complete result for TD1 datasets. Similar argument goes with the TD2 datasets with a less effect as only 3CE cannot be assigned a completed schedule for OH1 and OH6.

The performance of OH2, OH3 and OH4 on the modified datasets is the worst among all 6 heuristics. Reducing the number of classes by creating a combination of two-hour and one-hour classes produce a better performance under OH1, OH5 and OH6 because the complete schedules can be achieved.

Table 4-40 Characteristic of the dataset

Student group	Class size		Teacher			Study hour			Room	Total
	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	CH9	
Eng A - I		✓	✓			✓				3
Eng J - R		✓	✓	✓		✓				4
2ME	✓		✓		✓	✓				4
2CoE	✓			✓	✓	✓				4
2EE		✓	✓		✓					3
2IE			✓	✓		✓	✓			4
2MnE	✓		✓				✓			3
2CE							✓			1
2ChE		✓	✓				✓			3
3ME	✓				✓	✓	✓			4
3CoE					✓					1
3EE		✓	✓		✓	✓				4
3IE	✓	✓	✓	✓	✓	✓	✓			7
3MnE	✓	✓	✓	✓	✓	✓		✓		7
3CE	✓	✓		✓	✓	✓		✓		6
3ChE				✓						1
4ME		✓	✓	✓	✓	✓	✓	✓	✓	8
4CoE								✓		1
4EE		✓		✓	✓			✓		4
4IE		✓		✓	✓			✓		4
4MnE		✓	✓	✓	✓	✓		✓		6
4CE	✓				✓	✓		✓		4
4ChE		✓		✓	✓					3

4.9 Guideline

To apply the proposed prototype on any dataset, many parameters must be set such that the prototype will perform well in terms of the processing speed and the user satisfaction. Firstly, the datasets of unrelated classes must be divided into different trees in order to speed up the searching process. For example, the first year subjects to be offered to all students might get a higher priority to be allocated the timeslots and the rooms first before other classes. This requirement can be achieved by creating a search tree specifically for such classes. Once, the results of such classes are completed. The results can be loaded as a predefined class-time before starting the searching process for all other classes of each curriculum in each faculty. Furthermore, if the resources or the teachers are not overlapped with each other, then such curriculums can be processed in parallel. This way, the processing time can be reduced.

Second, the ordering heuristic to be used as a baseline is OH1. OH5 and OH6 show some good performances on achieving the best score from many datasets. However, the performance difference between OH1 and both OH5 and OH6 is minimal in terms of soft constraints. Furthermore, the OH1 always provides a better hard constraint violation performance. With the modified datasets, OH1 only fail 6 of 172 cases to produce a complete schedule, following by OH6 at 9. Thus, the first recommendation ordering heuristic is OH1. If the workload does not contain various class-size then OH6 can be applied.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The classroom timetabling system was developed and evaluated by the workload of the Faculty of Engineering, Prince of Songkla University. Depth-bounded Discrepancy search (DDS) is applied together with the heuristic and hard-soft constraints to guide the search to a good solution space. To adapt on changes of inputs, the proposed system allows the constraints as a system configuration to be set by the user. Sections 4.3 and 4.4 show that the proposed system can find a feasible solution for the datasets. The proposed system creates weekly timetables of all datasets in the case study before the result is displayed as a web-based format for each student group, each teacher and each room timetables. The constraints are classified into two groups depending on how important of the constraints to the resulting timetables. The proposed systems support 8 types of constraints from 5 categories which are also described in Sections 2.3 and 3.6.

The original university timetable, the first depth DDS and the second depth DDS results are compared in Section 4.1. The comparison shows that DDS can find a better solution. The conflicts are reduced into nearly zero. The remaining conflict came from limited resource cases including multiple teachers and after business hour cases. Section 4.3 shows that the first depth DDS can find a feasible solution. However, the second depth DDS can also further reduce the number of hard constraint violations. Since, the second depth DDS increases the number of paths to probe in each search tree, the processing time of the second depth DDS will be increased to complete the search for all datasets.

Section 4.5 presents that the objective model can manage the different priority of constraints. And, the results show that the time constraints are more effective to the search engine than that of the time-spread constraint because the time constraints are used as the guideline for selecting the area in the timetable to assign the timeslot for a class.

The different ordering heuristics are also applied to the search engine. Section 4.6 shows that the ordering heuristic, emphasizing on class size more than teacher workload and study-hour respectively, is better than other heuristics. However, ordering heuristics emphasizing on study-hour first are also providing adequate results.

The performance of the proposed prototype is presented in Section 4.7. While Section 4.8 and 4.9 analyze the results in order to create a guideline on how to apply the search with another dataset.

5.2 Future work

The proposed system is designed and developed as a sequential program. However, some parts of the process can be done simultaneously. Thus, parallel programming can be applied to improve the searching time of the system. For example, the score calculation in Figure 3-1 can be improved to calculate the score of each timeslot in parallel. DDS can also search simultaneously on different subtrees. However, the data model must be stored separately to avoid an error.

The inheritance scheme of Java object enables the system to allow additional constraints. For example, if the institute wants to add more constraints such as the distance between rooms. The constraint that focuses on reducing the distance between adjacent classes can be implemented and imported to the system.

REFERENCES

[1] K. Murray, T. Müller, and H. Rudová, "Modeling and solution of a complex university course timetabling problem," *Pract. Theory Autom. Timetabling VI*, vol. 3867, pp. 189–209, 2006.

[2] S. Chaiyasuwan and W. Tantithamphusit, "The Satisfaction and Expectation of Teachers and Staff Working in Prince of Songkla University to the Learning, Teaching and Examination," 2007.

[3] S. Chaiyasuwan and W. Tantithamphusit, "The Satisfaction and Expectation of Student in Prince of Songkla University to the Learning, Teaching and Examination Schedule Management," 2007.

[4] S. Chaiyasuwan and W. Tantithamphusit, "Operating Manual For Learning, teaching and examination time-table management," 2007.

[5] P. Ameen, "A Genetic Algorithm-based Timetable for Undergraduate Engineering Student, PSU," Prince of Songkla University, 2012.

[6] T. Walsh, "Depth-bounded Discrepancy Search," in *Proceedings of the Fifteenth International Joint Conference on Artifical Intelligence - Volume 2*, 1997, pp. 1388–1393.

[7] L. Di Gaspero, B. McCollum, and A. Schaerf, "The Second International Timetabling Competition (ITC-2007): Curriculum-based Course Timetabling (Track 3)," *Electr. Eng.*, no. Track 3, pp. 1–12, 2007.

[8] G. Lach and M. E. Lübbecke, "Curriculum based course timetabling: New solutions to Udine benchmark instances," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 255–272, 2012.

[9] D. Sánchez-Partida, "Modeling and solving a timetabling problem considering time windows and consecutive periods," *Lect. Notes ...*, vol. 5, pp. 25–32, 2013.

[10] S. A. MirHassani, "A computational approach to enhancing course timetabling with integer programming," *Appl. Math. Comput.*, vol. 175, no. 1, pp. 814–822, Apr. 2006.

[11] N. Boland, B. D. Hughes, L. T. G. Merlot, and P. J. Stuckey, "New integer linear programming approaches for course timetabling," *Comput. Oper. Res.*, vol. 35,

no. 7, pp. 2209–2233, Jul. 2008.

[12] S. M. Al-Yakoob and H. D. Sherali, "A mixed-integer programming approach to a class timetabling problem: A case study with gender policies and traffic considerations," *Eur. J. Oper. Res.*, vol. 180, no. 3, pp. 1028–1044, Aug. 2007.

[13] E. K. Burke, J. Mareček, A. J. Parkes, and H. Rudová, "A supernodal formulation of vertex colouring with applications in course timetabling," *Ann. Oper. Res.*, vol. 179, no. 1, pp. 105–130, Mar. 2010.

[14] M. Banbara, T. Soh, N. Tamura, K. Inoue, and T. Schaub, "for Course Timetabling," 2013.

[15] R. Lewis, "A survey of metaheuristic-based techniques for University Timetabling problems," *OR Spectr.*, vol. 30, no. 1, pp. 167–190, 2008.

[16] R. Bellio, S. Ceschia, L. Di Gaspero, A. Schaerf, and T. Urli, "Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem," *Comput. Oper. Res.*, vol. 65, pp. 83–92, Jan. 2016.

[17] N. Rangel-Valdez, J. O. Jasso-Luna, M. H. Rodriguez-Chavez, and G. Bujano-Guzman, "Practical relaxation of a special case of the Curriculum-Based Course Timetabling problem," *Prog. Artif. Intell.*, vol. 2, no. 4, pp. 237–248, 2014.

[18] A. Gunawan, K. M. Ng, and K. L. Poh, "A hybridized Lagrangian relaxation and simulated annealing method for the course timetabling problem," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3074–3088, Dec. 2012.

[19] J. A. Soria-Alcaraz, E. Özcan, J. Swan, G. Kendall, and M. Carpio, "Iterated local search using an add and delete hyper-heuristic for university course timetabling," *Appl. Soft Comput.*, vol. 40, pp. 581–593, Mar. 2016.

[20] Z. Lü and J. K. Hao, "Adaptive Tabu Search for course timetabling," *Eur. J. Oper. Res.*, vol. 200, no. 1, pp. 235–244, 2010.

[21] K. Nguyen, N. Dang, K. Trieu, and N. Tran, "Automating a real-world university timetabling problem with Tabu search algorithm," *2010 IEEE-RIVF Int. Conf. Comput. Commun. Technol. Res. Innov. Vis. Futur. RIVF 2010*, 2010.

[22] P. De Causmaecker, P. Demeester, and G. Vanden Berghe, "A decomposed metaheuristic approach for a real-world university timetabling problem," *Eur. J. Oper. Res.*, vol. 195, no. 1, pp. 307–318, 2009.

[23] R. P. Badoni, D. K. Gupta, and P. Mishra, "A new hybrid algorithm for university course timetabling problem using events based on groupings of students," *Comput. Ind. Eng.*, vol. 78, pp. 12–25, Dec. 2014.

[24] a Jain, D. Jain, and D. Chande, "Formulation of Genetic Algorithm to Generate Good Quality Course Timetable," *Int. J. Innov. ...*, vol. 1, no. 3, pp. 248–251, 2010.

[25] W. Chinnasri and N. Sureerattanan, "Comparison of performance between different selection strategies on genetic algorithm with course timetabling problem," *Adv. Manag. Sci. (ICAMS), 2010 IEEE Int. Conf.*, vol. 2, pp. 105–108, 2010.

[26] P. Pongcharoen, W. Promtet, P. Yenradee, and C. Hicks, "Stochastic Optimisation Timetabling Tool for university course scheduling," *Int. J. Prod. Econ.*, vol. 112, no. 2, pp. 903–918, 2008.

[27] C. Nothegger, A. Mayer, A. Chwatal, and G. R. Raidl, "Solving the post enrolment course timetabling problem by ant colony optimization," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 325–339, Feb. 2012.

[28] A. L. Bolaji, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "University course timetabling using hybridized artificial bee colony with hill climbing optimizer," *J. Comput. Sci.*, vol. 5, no. 5, pp. 809–818, Sep. 2014.

[29] M. A. Al-Betar and A. T. Khader, "A harmony search algorithm for university course timetabling," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 3–31, Jul. 2010.

[30] N. R. Sabar, M. Ayob, G. Kendall, and R. Qu, "A honey-bee mating optimization algorithm for educational timetabling problems," *Eur. J. Oper. Res.*, vol. 216, no. 3, pp. 533–543, Feb. 2012.

[31] M. S. Kohshori and M. S. Abadeh, "Hybrid Genetic Algorithms for University Course Timetabling," *Int. J. Comput. Sci. Issues*, vol. 9, no. 2, pp. 446–455, 2012.

[32] R. Bellio, L. Di Gaspero, and A. Schaerf, "Design and statistical analysis of a hybrid local search algorithm for Course Timetabling," *J. Sched.*, vol. 15, no. 1, pp. 49–61, 2012.

[33] G. Post, L. Di Gaspero, J. H. Kingston, B. McCollum, and A. Schaerf, "The Third International Timetabling Competition," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 69–75, 2016.

[34] A. Bonutti, F. De Cesco, L. Di Gaspero, and A. Schaerf, "Benchmarking curriculum-based course timetabling: Formulations, data formats, instances, validation, visualization, and results," *Ann. Oper. Res.*, vol. 194, no. 1, pp. 59–70, 2012.

[35] T. Müller and H. Rudová, "Real-life curriculum-based timetabling with elective courses and course sections," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 153–170, 2016.

[36] T. Müller and H. Rudová, "Real-life curriculum-based timetabling with elective courses and course sections," *Ann. Oper. Res.*, vol. 239, no. 1, pp. 153–170, Jun. 2014.

[37] A. Schaerf and L. Di Gaspero, "Measurability and Reproducibility in Timetabling Research: Discussion and Proposals," *ractice Theory Autom. Timetabling VI, 6th Int. Conf. PATAT 2006, Brno, Czech Republic, August 30 - Sept. 1, 2006, Revis. Sel. Pap.*, vol. 3867, pp. 40–49, 2007.

[38] B. McCollum, "A Perspective on Bridging the Gap Between Theory and Practice in University Timetabling," in *Practice and Theory of Automated Timetabling VI: 6th International Conference, PATAT 2006 Brno, Czech Republic, August 30--September 1, 2006 Revised Selected Papers*, E. K. Burke and H. Rudová, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 3–23.

[39] S. Vasupongayya and S. h. Chiang, "Multi-Objective Models for Scheduling Jobs on Parallel Computer Systems," in *2006 IEEE International Conference on Cluster Computing*, 2006, pp. 1–6.

[40] "Java official website." [Online]. Available: https://www.java.com/.

[41] "MySQL official website." [Online]. Available: https://www.mysql.com/.

[42] "CodeIgniter official website." [Online]. Available: https://www.codeigniter.com/.

[43] "AngularJS official website." [Online]. Available: https://angularjs.org/.

[44] "Bootstrap official website." [Online]. Available: http://getbootstrap.com/.

[45] "Apache HTTP server project official website." [Online]. Available: https://httpd.apache.org/.

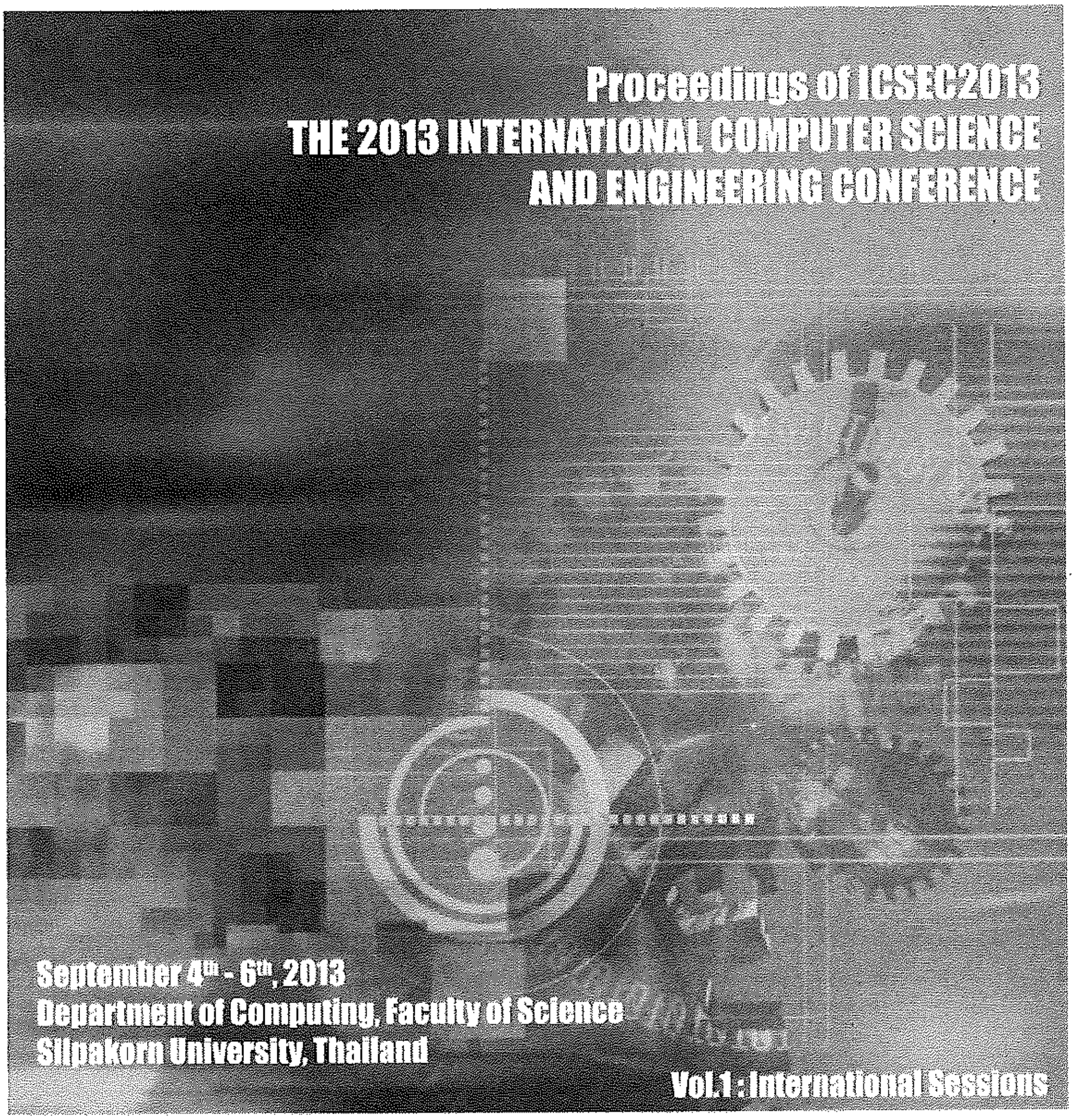
APPENDIX

APPENDIX A
PUBLICATION

1) W. Sitthirit, S. Vasupongayya, "Applying a mixed objective model in a university timetabling solution searching technique", in Proceedings of *2013 International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2013

17th ICSEC

International Computer Science
and Engineering Conference 2013



2013 International Computer Science and Engineering Conference (ICSEC): ICSEC 2013 English Track Full Papers

Applying a Mixed Objective Model in a University Timetabling Solution Searching Technique

W. Sitthirit[1] S. Vasupongayya[2]
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University,
Hat Yai, Songkhla 90112, Thailand
E-mail: kornsitti@gmail.com[1], vsangsur@coe.psu.ac.th[2]

Abstract—**Course timetable is a problem that can be solved using a search technique. The task is usually required in all educational institutes. Even though several literature's had been proposed for the problem, the practical solution is still required a lot of human modifications due to a list of specific requirements and the different characteristics of each institution. This work aims to develop a flexible model to guide the search technique namely dept-bounded discrepancy search to solve timetabling problems. The experiments are done using the real data set of the Faculty of Engineering at Prince of Songkla University. The results show that the objective model can guide the search to a solution that satisfies all hard constraints with a minimal number of soft-constraints violations. However, the order of the soft constraints considered by the search technique can affect the number of soft-constraints violations.**

Keywords-university, depth-bounded discrepancy search, constraint, priority, timetable, scheduling

I. Introduction

Course timetabling problems are considered as an example of classic constraint optimization problems. Several works have been proposed in the past [1][2][3][5][7]. However, the results or processes of these solutions still require human adjustment processes due to the specific requirements of each institute and the differences in characteristics of the institute. Search techniques are the most common techniques to solve constraint optimization problem. This work aims to develop a flexible model to guide a search technique namely Dept-bounded Discrepancy Search (DDS) [18] to solve timetabling problems. The results of this work will be the first step of developing a flexible objective model for the proposed framework to solve course timetabling problems that can be turned to fit specific requirements of each institute.

A course timetabling problem is a process of allocating courses to available and suitable rooms. Unless the institution uses a fit course schedule, the institution will have to complete this process every semester. The process will have a large amount of data including courses, student groups, teachers, classrooms, and a list of associate requirements. For example, some courses have limited numbers of students; some courses must be in specific rooms; some courses must be in specific time of days. Only solving common constraints is not practical enough. Thus, the scheduling framework must be flexible enough for a specific need of each institute to be realistic. However, the specific needs cannot be foreseen. To solve such complicated problem, a search technique is employed. Since, the search technique goes through the space of solutions to find a suitable solution. By guiding the search to value a list of requirements will create a suitable framework for such tasks.

There are two types of search techniques which are the search algorithm that works on a space of total solutions such as local search [8][21][22] and the search algorithm that works on a space of partial solutions [18][19]. Since the number of rooms and the number of lecturers are limited, the earlier courses that are assigning the lecturer time slot and the rooms can affect the later courses. Thus, the searching technique that works on a space of partial solutions to produce a piece of solution will be suitable for such task. Since the search space can be very large, getting to see the potential of each area of the search space early will help the search to find a good area to be explored. Thus, the discrepancy based search namely dept-bounded discrepancy search that forces the search to probe into each subtree before moves to the promising subtree is selected as the main searching technique in this framework.

To guide the search, each requirement of the institute can be defined as a constraint. This way, any specific requirement can be satisfied by defining as a constraint. Thus, the proposed framework can be practical for solving real course timetabling problems. However, each constraint can have different effects on the results. Thus, constraints may need to be assigned a different priority. For example, some constraints or requirements must be satisfied in all situations, while some constraints or requirements are preferred if it is possible to satisfy. Therefore, the constraints must be specified their priority as well. Since each institute can have several constraints, the organization of these constraints must be designed. Even though the constraint priority depends on the user requirements, different organization of the constraints will cause different effects on different set of workloads.

The organization of the constraints in this work is referred to as an objective model. The objective model guides the search to a specific area of the search space because the object model will evaluate the 'goodness' of solutions in each area. Thus, if the search finds a good solution then the search is continued otherwise the search will move to a different area. This paper aims to study the impact of a mix objective model. In order words, the paper aims to study the impact of different way to organize the constraints.

The remaining of this paper is organized as follow. In Section II, the literature reviews are given. Section III described the experimental settings. Section IV presents the experimental results and discussions. Section V gives the conclusion.

II. Literature Reviews

This section provides information regarding the researches on algorithms or methods to solve course timetabling problems in Section A. Next, the requirements of the input data set at the Faculty of Engineering at Prince of Songkla University are described in Section B. Then, the objective model concepts are described in Section C. The dept-bounded discrepancy search algorithm is explained in Section D.

A. *Solutions to course timetabling problems*

There are several works on methods to solve course timetabling problems. The survey on timetabling methods for high school and university level is given in [1]. The early method that often uses for solving a course timetabling problem is mathematics models such as integer linear programming [2][3][16] or reduce graph coloring [4]. Later, many searching techniques [5] are employed to solve the problem due to the increasing amount of data. The next set of methods are machine learning based methods such as genetics algorithm [6] [7], local search [8] [9], expert System [10], distributed computing [11], and data mining techniques [12]. In [13] several methods are compared including ant colony optimization, genetics algorithm, iterated local search, simulated annealing and Tabu search. The results showed that each method was good on some sets of data. No method is good for all data sets due to the different characteristics of the data set.

Thus, this work aims to propose a framework with a flexible objective model that can be adapted to a different set of requirements without a need for low level changing of the underlining method. Therefore, this work selects a search method to be the core and the objective model conducted from the specific problem requirements to guide the search to a good solution in the space of partial solutions.

B. *Faculty of Engineering, Prince of Songkla University*

According to [13] the data set of Faculty of Engineering at Prince of Songkla University is considered a large instance. Typically, there are more than 600 subjects for students to register in the first semester of each year [14]. There are 12 undergraduate programs, 11 master level programs, and 8 doctoral programs. These curriculums are from 7 different departments. Each department has different characteristic of requirements. Several lecturers are teaching for several curriculums. There are approximately 170 lecturers with approximately 4,000 active students.

Currently, the course timetabling process at Prince of Songkla University is a semi-automatic process [15]. That is, the Registrar's Division will forward the previous year schedules to every department under Faculty of Engineering. Each department has to specify any change such as teacher changes, subject ID changes, and new subjects. Then, the changes must be returned to the Registrar's Division. Finally, the Registrar's Division uses the collected information to modify the previous year schedule for the new academic year. The process is done partly by human and a simple program to show the conflicts. However, the Registrar's Division final schedules are known to contain a lot of conflicts such as room conflicts, time conflicts, and examination time conflicts.

There are a few attempts at developing a program to solve such tasks. In [16], a set of hard and soft constraints is created to be used together with a simple greedy algorithm. A problem specific knowledge is created as a set of heuristics to solve the course timetabling for the Department of Computer Engineering at Prince of Songkla University. The experimental results in [16] showed that their algorithm could find a schedule without any time conflict, any teacher conflict, or any room conflict. However, the greedy method relies too much on the heuristics. Therefore, it is difficult to apply such technique on a large scale problem. In [20], a genetic algorithm is employed to provide a schedule for the Faculty of Engineering at Prince of Songkla University. This work also employs the hard and soft constraints. The experimental results show that their hard constraint violations can be reduced at the expenses of increasing the soft constraint violations.

This work will use the data set from the Faculty of Engineering at Prince of Songkla University as the sample problem set to evaluate the objective model studied in this work. Due to its large scale and variation of requirement, this data set can draw the problems or issues of each objective model to light.

C. *Objective model*

The requirements of each institute can be transformed into a set of constraints. However, these constraints must be put together as a unit called an objective model. In previous work [17], there are three objective models proposed. Each of which has a specific property. The first model is called Lexical model denoted by *Lexical*(A→B). Under this model constraint A is dominated constraint B. That is, a solution that provides the best performance on constraint A is considered good. Even though this solution may provides the worst performance on constraint B, the solution is still considered a better solution because A is more important than B. The second model is called Ordered-Tradeoff denoted by *Tradeoff(A→B)*. Under this model, the slightly worse performance of constraint B is acceptable as long as the improvement of constraint A is significant. The degree of improvement is calculated as the ratio of improvement comparing with the first solution found. Thus, this second model does not allow constraint A to dominate the constraint B as much as the first model. The last model is called Equal-Tradeoff Model denoted by *Tradeoff(A:B)*. Under this model, the solution with the most improvement either in A or in B is considered good because A and B are equal. However, the improvement that comes with too much degradation of the other will be considered "no good".

In this work, the first and the second models are applied. Thus, it is the origin of a mixed model meaning there are two models in this work. The first model (Lexical) is applied for the hard constraints which cannot be violated. The second model (Ordered-Tradeoff) is applied for the soft constraints which can be violated if necessary. The last model (Equal-Tradeoff) is not applied in this work because typical requirements are of different priority. However, an experimental of equal priority of soft constraints is also conducted for performance evaluation purposes.

D. Dept-bounded discrepancy search

Dept-bounded Discrepancy Search (DDS) [18] is a search technique in the space of partial solution. The search space can be viewed as a search tree. For example, Figure 1 shows the search tree of eight solutions. The search will start from the root to the leaf which is called a path or a solution. Once, the search reaches the leaf then a candidate solution is found. The objective model will be used to calculate the score of the solution. The search continues and the score will be calculated every time that the search reaches the leaf. If the new solution is better than the old solution then the new solution is recorded.

DDS will probe into each subtree very quick because DDS will visit a path of each subtree. For example, Figure 1 shows the order of leaf to be discovered by the DDS. The first leaf is the right-most path. The next path is the 5rh path from the right or a path from a different subtree with the first discovered solution. This way, DDS gives a high priority on the nodes closed to the root than that closed to the leaf. This property is suitable with the course timetabling problem because the earlier course assignments can have a huge impact on the later courses. Furthermore, the objective model can be calculated on a partial solution for the Lexical model. Thus, the pruning technique can be used to avoid searching in the area containing solutions that are not better than the current solution found so far. In this work, DDS will be used as the main search technique in our search engine.

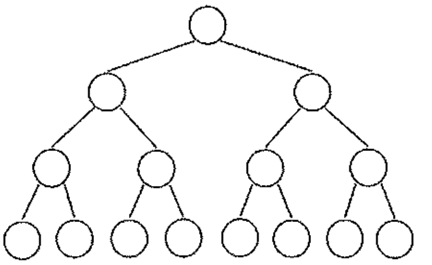


Figure 1. Dept-bounded Discrepancy Search.

III. EXPERIMENTAL SETTINGS

In this section, the details of the constraints in the experiments are described in Section A. In addition, the problem domain heuristics used to define the set of constraints are also explained. The full description of the algorithm is given in Section B. Finally, the workloads and measurements are presented in Section C.

A. Constraints

As mentioned above, some requirements can be significantly more important than others. Thus, the constraints must be classified so that the system can find a suitable solution to the problem at hands. In this work, there are two types of constraints. There are hard constraints and soft constraints. The hard constraints are considered to be more important than the soft constraints. For example, each subject must be assigned a large enough room otherwise the subject cannot be taught in the room. Thus, the constraints that cannot be violated will be defined as hard constraints. The soft constraints, on the other hands, are the constraints that can be violated if necessary. For example, the schedule of each student should not exceed five consecutive periods in each day. This constraint may not be possible for some students such as those that still have to re-take some subjects.

In this work, the sources of hard constraints are collected from the problems encountered by the Faculty of Engineering staffs at Prince of Songkla University. The most important constraint for course timetabling problem is that all subjects must be assigned rooms and time slots. The assignments must not create any conflict, especially the conflict that prevents the students to register a set of subjects according to their curriculums. Furthermore, the conflict on teachers must be avoided in order to run the current curriculum smoothly.

The list of hard constraints includes H1, the students must be able to enroll in subjects according to their curriculum; H2, each lecturer must be assigned a single subject during each time slot; H3, each room must be assigned to a single subject during each time slot; H4, the class time should be assigned during regular business hours; H5, the drawing class should be finished before 8:00PM; H6, the laboratory subjects should be finished before 6:00PM; H7, no class should be scheduled on Saturday or Sunday; H8, no class should be scheduled at noon.

The soft constraints, in this work, are focusing on requirements that are not as significant as the hard constraints. The list of soft constraints includes S1 the schedule of each student according to his/her curriculum should not exceed 6 hours each day; S2, the maximum free hours of each student should not exceed 3 hours; S3, the same subject must not be assigned in the adjacent days; S4 the laboratory subjects should be assigned after 2:00PM; S5 the same subject should be assigned the same time slots on each day.

B. Algorithms

In this section, the framework proposed to solve the course timetabling problems is described. Figure 2 shows the overall structure of the proposed framework. The main framework includes two parts which are the search engine and the objective model. There are two sets of input to the framework which are the information related to the course timetabling and the information related to the constraints or the specific requirements of the framework. The course timetabling related information such as the number of available room and its description, the subject information, the lecturer information, the curriculum information, will be used by the search engine to find the solution. The constraints will be used in the objective model in order to guide the search engine.

The search engine employs DDS as the main method to search in the space of partial solutions. The partial solution is the partial list of assignments including a classroom, a subject, a time slot, a list of lecturers, and a list of student groups. The complete solutions are the completed list of assignments. Since the DDS will search into the left path of each subtree. The order of subjects to be considered will affect the final results. That is, the earlier subject will be assigned the rooms before the later subjects. Thus, the later subject choices are limited by such decisions.

In this work, a simple subject ordering heuristics resulting from the previous work [16] is employed. That is, the subjects should start from the freshman level classes because they are a few classes. However, this group of students creates a lot of time conflicts due to their curriculum because the freshmen take a lot of basic sciences and social science subjects taught by other faculties. Then, the next set of subjects is the sophomore, junior and senior level subjects. For subjects with in the same level, the laboratory subjects are considered to be higher priority than regular subjects because they have specific requirements of the class room and class time.

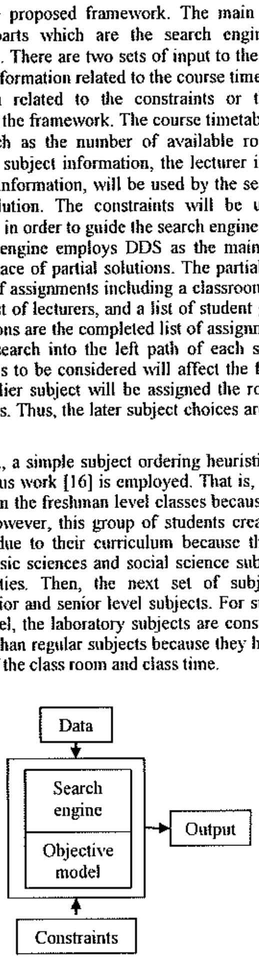


Figure 2. Overall structure of the proposed framework

C. *Workload and Measurements*

To study the effects of the objective models and several design choices, the framework is evaluated using the data set of the first semester of 2012 academic year of the Faculty of Engineering at Prince of Songkla University. There are 188 subjects from Faculty of Engineering includes 34 laboratory subjects, 144 lecture subjects and 10 subjects that has both lecture and laboratory class. There are 14 subjects from other faculties. 27.23% of subjects are taught by multiple lecturers. There are 158 lecturers. There are 74 rooms from 10 buildings in Faculty of Engineering which are available for classes. There are 74 registration patterns from 40 groups of students in total. The number of constraint violations is reported and the detail analysis of each violation is studied to find the effects of each constraint.

Three experimental settings are organized in this work. First, the results of the simple mixed objective model as described in Section II are compared against that of the current schedules used by the university in the first semester of 2012 academic year which will be denoted by US. The second experiment is focusing on the ordering of the soft constraints in the ordered tradeoff objective model of soft constraints. The last experiment aims to compare the performances of all objective models.

The simple version mixed objective models (SMO) give equal priority to all eight hard constraints and equal priority for all six soft constraints. That is, the total number of violations of all hard constraints is the scoring of the hard constraints. Then, the total number of violations of all soft constraints is the scoring of the soft constraints. Both values will be put into a lexical objective model. That is, the solution with a smaller number of hard constraint violations is considered a better solution. If two schedules provided the same number of hard constraint violations then the schedule with the smaller number of soft constraint violations is considered a better solution. If two schedules provide exact same number of both violations, the first schedule discovered by the scheduler is considered better because it follows the domain knowledge heuristic.

For the first mixed objective model (MO1), in this work, the soft constraints are organized in the ordered tradeoff objective model by first ordering S4 then S3 and S5 then S1 and S2. Thus, for each solution the number of S4 violations must be calculated; the number of S3 and S5 violations must be calculated; the number of S1 and S2 violations must be calculated; the number of hard constraints must be calculated. Then, the improvement or degradation ratio of each set of constraints is calculated. For each pair of schedules, the schedule with the least number of hard constraints is considered a better schedule. If two schedules provide the same number of hard constraint violations then the schedule with the most improved ratio will be considered a better solution. However, the improvement ratio of S4 is considered more important than the improvement ratio of S3 and S5 which is in turn considered more important than the improvement ratio of S1 and S2. For the second mixed objective model (MO2), in this work, the order of the soft constraints organization in the ordered tradeoff objective model is the reverse of that under the MO1.

The S1 and S2 constraints are grouped together due to its characteristic of the numbers of consecutive learning and free hours of students. The S3 and S5 constraints are grouped together due to its characteristics of the time slot of each subject.

IV. Experimental Results

Table 1 shows the number of hard constraint violations under each objective model and the schedule currently used at the university. Table 2 shows the number of soft constraint violations under each objective model. Table 3 shows the number of hard constraint violations under the schedules used by the university when excluding all the subjects that the scheduler is not responsible for such as the subjects taught by lecturers outside the Faculty of Engineering. The results of our proposed objective model are not shown in Table 3 because they are all zero. Meaning, the proposed objective model can find a solution that avoids all the hard constraint violations.

According to the performances of the current university schedules (denoted as US in Table 1 and Table 2), there are a lot of hard and soft constraint violations, especially the H3, S3 and S5. That is, the current university schedules make a lot of mistakes in assigning the non-empty room to another subject. The resulting schedules of each subject are also assigned on the adjacent days and at a different time slot. Table 3 presents the number of hard constraint violations when removing the violations caused by the subjects taught by other faculties. The number of violations under H2, H4, H7 and H8 of the current university schedule improves due to the removing of conflicts caused by the subjects of different faculties.

The simple mixed objective model performances (denoted as SMO in Table 1 and Table 2), however, are better than that of the current university schedule under all cases. Especially, the number of hard constraints when the conflicts caused by the subjects taught by other faculties are removed is now zero. Meaning, the SMO can find a solution that does not violate any hard constraint.

Table 1 The hard constraints violations

	US	SMO	MO1	MO2
H1	56	6	6	6
H2	90	77	71	64
H3	136	0	0	0
H4	86	60	58	60
H5	0	0	0	0
H6	2	0	0	0
H7	12	9	9	9
H8	14	5	5	5

Table 2 The soft constraints violations

	US	SMO	MO1	MO2
S1	59	54	60	**47**
S2	98	30	35	**26**
S3	219	153	**145**	185
S4	**18**	12	**5**	33
S5	345	**256**	283	306

Table 3. The hard constraint violations ignoring subjects taught by other faculty under current schedule used at the university

H1	H2	H3	H4	H5	H6	H7	H8
50	21	136	30	0	2	3	9

For the performances of other mixed objective models (denoted as MO1 and MO2 in Table 1 and Table 2), the performances are similar to that under SMO. The number of hard constraint violations is considered similar among all three objective models. For the number of soft constraint violations, all three mixed objective models produce better performances than that of the current university schedule.

To analyze the details of soft constraint violations, MO1 provides a good S4, S3 and S5 performances comparing with MO2 because MO1 orders the constraints as S4, {S3, S5}, {S1, S2}. On the other hand, MO2 provides a good S1 and S2 performances because MO2 orders the constraints as {S1, S2}, {S3, S5}, S4. As a result, MO2 provides the worst S4 performance because MO2 gives less priority to the S4 constraint. Similarly, MO1 provides the worst S1 performance because MO1 gives less priority to the S1 constraint. These events confirm that the objective model does proceed according to its ordering.

V. CONCLUSION

This paper presents the impact of mixed objective models on the data set of 2012 academic year of the Faculty of Engineering at Prince of Songkla University. Constraints or requirements are organized into a mixed objective model. Three different mixed objective models are proposed in this work to test the impact of each model. The first model (SMO) tests only the lexical idea between the hard constraints and the soft constraints. The experimental results show that the SMO can find a solution that does not violate any hard constraint. The two mixed objective model (MO1 and MO2) tests the idea of using the ordered tradeoff idea among soft constraints. The experimental results show that the ordered tradeoff among soft constraints provides the solution strictly according to the objectives defined.

The future works will focus on the user interface to define a set of constraints to the framework so that these constraints can be transformed into an appropriated objective model. Furthermore, the classes of constraints in the university timetabling problem will be defined and studied their impacts on the search engine. Once the framework is done, it will be evaluated by the perspective users and on several data sets.

ACKNOWLEDGEMENT

This research is supported by Prince of Songkla University under ENG550405S and PRPM ID 12034. The authors also would like to thanks the Center of Network Research at Prince of Songkla University for the facility support.

REFERENCES

[1] A. Schaerf, A survey of automated timetabling on Artificial Intelligence Review. vol. 13 no. 2, 1999 . pp. 87-127.

[2] A. Tripathy, School Timetabling A Case in Large Binary Integer Programing on Management Science, vol. 30 no. 12, 1984, pp. 1473-1489.

[3] A. Wasfy & F.A. Aloul, "Solving the University Class Scheduling Problem Using Advanced ILP Techniques", in proceeding of 4th IEEE GCC Conference, Bahrain, November 2007.

[4] D. De Werra. An introduction to timetabling on European Journal of Operational Research, vol. 19, 1985, pp. 151-162.

[5] I. Berada, J.A. Ferland & P. Michelon, A multi-objective approach to nurse scheduling with both hard and soft constraints on Socio-Economic Planning Science, vol. 30 no. 20, 1996, pp. 183-193.

[6] U. Aickelin & K. Downsland, Exploting problem strcture in genetic algorithm approach to a nurse rostering problem on Journal of Scheduling, vol.3 no. 3, 2001, pp. 139-153.

[7] J.W. Herrmann & C.-Y Lee. Solving a Class Scheduling Problem with a Genetic Aligorithm on ORSA JOURNAL ON COMPUTING, vol.7 no. 4, 1995, pp. 443-452.

[8] A Hertz. Tabu search for Large Scale Timetabling Problems on European Journal of Operational Research, vol.54 no. 1, 1991. pp. 186-193.

[9] E.L. Mooney, R.L. Rardin, & W.J. Parmenter. Large-scale Classroom Schedulin on IIE Transactions. vol. 28 no. 5, 1996, pp. 369-378.

[10] J.G. Chen & T.W. Yeung. Hybrid expert-system approach to nurse scheduling on Computer in Nursing. vol. 11, 1993. pp. 183-190.

[11] M. Dimopoulou & P. Miliotis, An automate University Course Timetabling System Developed in Distributed Environment on European Journal of Operational Research, vol. 153 no. 1, 2004, pp. 136-147.

[12] W. Smith. Applying Data mining to Scheduling Course at a University on Communications of the Association for Information Systems. vol. 16, 2005, pp. 463-474.

[13] E. Burke, P. De Causmaecker. "A Comparison of the Performance of Different Metaheuristics on the Timetabling Problem", in proceeding of 4th International the Conference on Practice and Theory of Automated Timetabling, 2002, 115-119.

[14] Faculty of Engineering. Faculty of Engineering Annual Report 2008-2009 Internet: http://phoenix.eng.psu.ac.th/qa/P&Q/contents/report_eng/ENG53.pdf, [Oct 25. 2012].

[15] N. Prayoonwong, Department of Computer Engineering. Personal Interview. March 13. 2012.

[16] W. Sitthirit & S. Vasupongayya, "Classroom Time Table Management System", in proceeding of OARIC2011, Phuket, Thailand, April 25-27, 2011.

[17] S. Vasupongayya & S.-H. Chiang. "Multi-Objective Models for Scheduling Jobs on Parallel Computer Systems", in proceeding of Cluster Computing. 2006 IEEE International Conference on, pp.1-6, 25-28 Sept. 2006

[18] T. Walsh, "Dept-bounded Discrepancy Search", International Joint Conference on Artificial Intelligence. vol. 2. 1997, pp. 1388-1393.

[19] W.D. Harvey , M.L. Ginsberg, "Limited Discrepancy Search", in proceeding of IJCAI'95, pp. 607-613, 1995.

[20] P. Ameen, "A Genetic Algorithm-based Timetable for Undergraduate Engineering Student, PSU", Master Thesis, Faculty of Engineering, Prince of Songkla University, 2012.

[21] H.Y. Al Tarawneh. M. Ayob. "Using Tabu search with multi-neighborhood structures to solve University Course Timetable UKM case study (faculty of engineering)", in proceeding of 3rd Data Mining and Optimization (DMO), pp.208-212, June 2011.

[22] E.K. Burke, G. Kendall, E. Soubeiga, A Tabu-Search Hyperheuristic for Timetabling and Rostering on Journal of Heuristics, vol. 9 no. 6, 2003, pp. 451-470.

APPENDIX B
THE DATABASE

This section describes the database structure of the proposed prototype which stores two groups of data including timetable data and searching data. Figure B-1 shows database structure of timetable data which stores the input/output of the system.

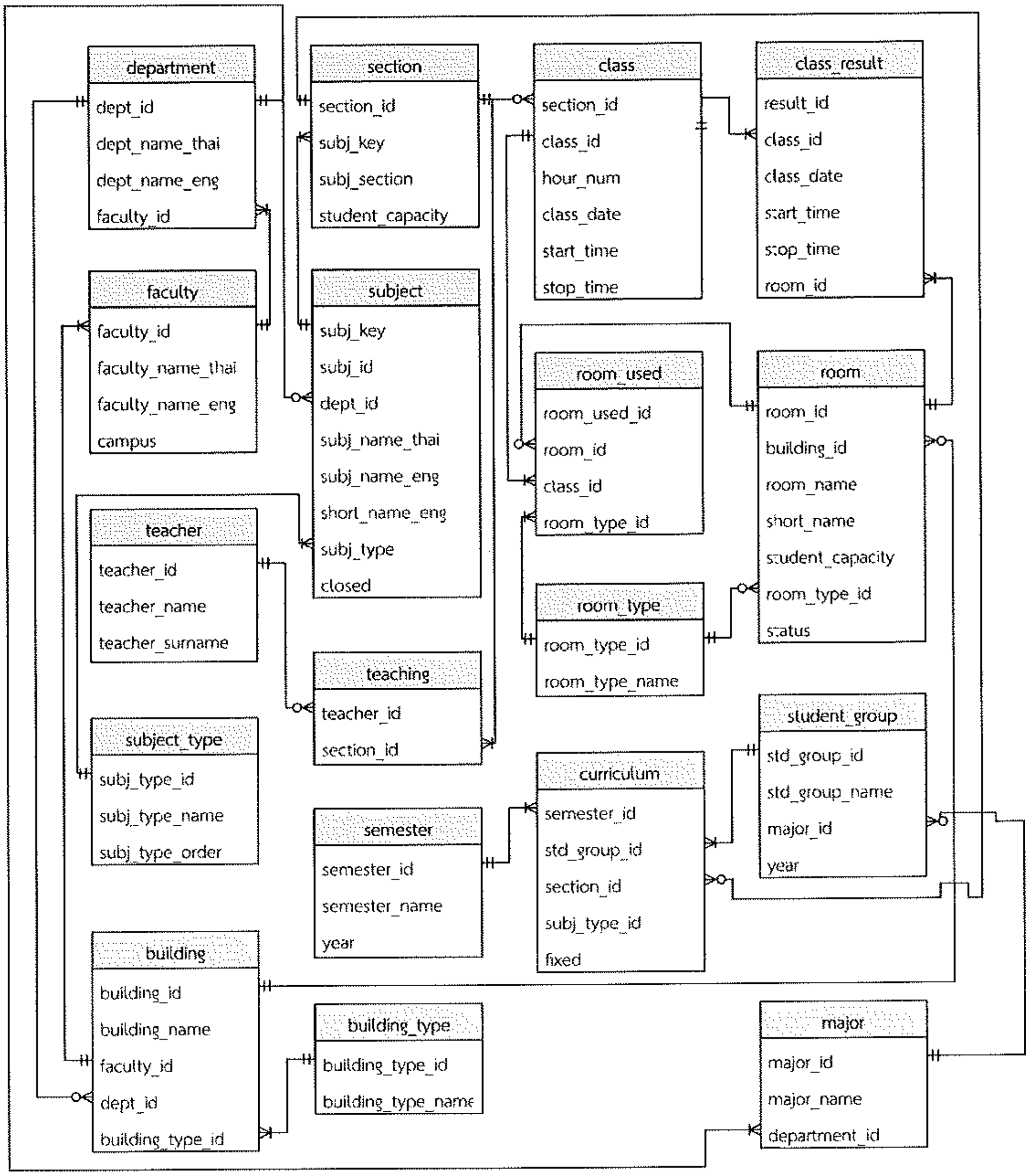


Figure B-1 The database structure for the timetable data

The details of each table in the timetable data are described follows.

1) class: This table stores the class information. The table fields are described follows.
   - class_id is the index of class table in an integer format.
   - section_id is the index of section table in an integer format.
   - hour_num is the number of required timeslots, 30 minutes per timeslot, in an integer format.
   - class_date is the date of the class in an integer format. The value is between 1 to 7 represents Monday to Sunday.
   - start_time is the start timeslot of the class in an integer format. The value is between 0 to 31 represents 7:00 AM to 10:30 PM.
   - stop_time is the end timeslot of the class in an integer format. The value is between 1 to 32 represents 7:20 AM to 11:00 PM.
2) section: This table stores the section information. The table fields are described follows.
   - section_id is the index of section table in an integer format.
   - subj_key is the index of subject table in a varchar format with seven digits.
   - subj_section is the section number in a varchar format with two digits.
   - student_capacity is the class capacity in an integer format.
3) subject: This table stores the subject information. The table fields are described follows.
   - subj_key is the index of subject table in a varchar format with seven digits.
   - subj_id is the subject code in a varchar format with seven digits.
   - dept_id is the index of department table in a varchar format with three digits.
   - subj_name_thai is the subject name in Thai language with a text format.
   - subj_name_eng the subject name in English language with a text format.
   - short_name_eng is the short name of the subject in English language with a text format.
   - subj_type is the index of subject_type table in an integer format.

- closed is the status of the subjects in an integer format. The value can be 0 or 1 represents inactive and active, respectively.

4) subject_type: This table stores the subject types for the subjects. The table fields are described follows.
   - subj_type_id is the index of subject_type table in an integer format.
   - subj_type_name is the subject type name in a text format.
   - subj_type_order is the subject type ordering in an integer format.
5) class_result: This table stores the results from the search engine for each class. The table fields are described follows.
   - result_id is the index of result table in an integer format.
   - class_id is the index of class table in an integer format.
   - class_date is the date of the class in an integer format. The value is between 1 to 7 represents Monday to Sunday.
   - start_time is the start timeslot of the class in an integer format. The value is between 0 to 31 represents 7:00 AM to 10:30 PM.
   - stop_time is the end timeslot of the class in an integer format. The value is between 1 to 32 represents 7:20 AM to 11:00 PM.
   - room_id is the index of room table in a varchar format with five digits.
6) Room: This table stores the room information. The table fields are described follows.
   - room_id is the index of room table in a varchar format with five digits.
   - building_id is the index of building table in a varchar format with four digits
   - room_name is the room name in a text format.
   - short_name is a short name of the room in a text format.
   - student_capacity is a capacity of the room in an integer format.
   - room_type_id is the index of room_type table in an integer format.
   - status is the availability of the room in an integer format. The value can be 0 or 1 represents inactive and active, respectively.
7) room_type: This table stores the room types for the rooms. The table fields are described follows.

- room_type_id is the index of room_type table in an integer format.
- room_type_name is the room type description in a text format.

8) building: This table stores the building information. The table fields are described follows.
    - building_id is the index of building table in a varchar format with three digits.
    - building_name is the building name in a text format.
    - faculty_id is the index of faculty table in a varchar format with two digits.
    - dept_id is the index of department table in a varchar format with three digits.
    - building_type_id is the index of building_type table in an integer format.
9) room_used: This table stores the required room type for the classes. The table fields are described follows.
    - room_used_id is the index of room_used table in an integer format.
    - class_id is the index of class table in an integer format.
    - room_type_id is the index of room_type table in an integer format.
    - room_id is the index of room table in a varchar format with five digits.
10) teacher: This table stores the teacher information. The table fields are described follows.
    - teacher_id is the index of teacher table in a varchar format with five digits.
    - teacher_name is the teacher name in a text format.
    - teacher_surname is the teacher surname in a text format.
11) teaching: This table stores the teacher workload which is the relation between teacher and section table. The table fields are described follows.
    - teacher_id is the index of teacher table in a varchar format with five digits.
    - section_id is the index of section_table in an integer format.
12) student_group: This table stores the student group information. The table fields are described follows.
    - std_group_id is the index of student_group table in an integer format.
    - std_group_name is the student group name in a text format.

- major_id is the index of major table in an integer format.
- year is the year of the student groups in an integer format.

13) semester: This table stores the semester information. The table fields are described follows

- semester_id is the index of semester table in an integer format.
- semester_name is the semester description in a text format.
- year is year of the semester in an integer format.

14) curriculum: This table stores the curriculum for each student group. The table fields are described follows.

- semester_id is the index of semester table in an integer format.
- std_group_id is the index of student_group table in an integer format.
- section_id is the index of section table in an integer format.
- subj_type_id is the index of subject_type table in an integer format.
- fixed is the status of the section in an integer format. The value can be 0, 1 or 2 represents unallocated, preloaded and allocated, respectively.

15) major: This table stores the major information. The table fields are described follows.

- major_id is the index of major table in an integer format.
- major_name is the major description in a text format.
- department_id is the index of the department in a varchar format with three digits.

16) department: This table stores the department information. The table fields are described follows.

- dept_id is the index of department table in a varchar format with three digits.
- dept_name_thai is the department name in Thai language with a text format.
- depth_name_eng is the department name in English language with a text format.
- faculty_id is the index of faculty table.in a varchar format with two digits.

17) faculty: This table stores the faculty information. The table fields are described follows.

- faculty_id is the index of faculty table in a varchar format with two digits.
- faculty_name_thai is the faculty name in Thai language with a text format.
- faculty_name_eng is the faculty name in English language with a text format.
- campus is the campus where the faculty places in a text format.

Figure B-2 shows database structure of searching data which stores the system configurations and the results. The details of each table in the searching data are described follows.

1) result: This table stores the result information. The table fields are described follows.

- result_id is the index of result table in an integer format.
- result_heuristic is the dataset order used in the search engine while produces the results. The value is in text format. The text string is composed of the dataset type and the dataset list. For example, to define the dataset in the student group type with two student groups, the text should be "dataset_type_id: std_group_id,std_group_id".
- result_start_time is the time when the user starts the search in datetime format.
- result_end_time is the time when finish the search in datetime format.
- result_score is the score of the result in text format.
- result_finish is the completeness of the result in an integer format. The value can be 0 or 1 represents uncompleted and completed, respectively.
- unallocated_class is the list of unallocated classes in a text format.
- result_description is the description of the result in a text format.

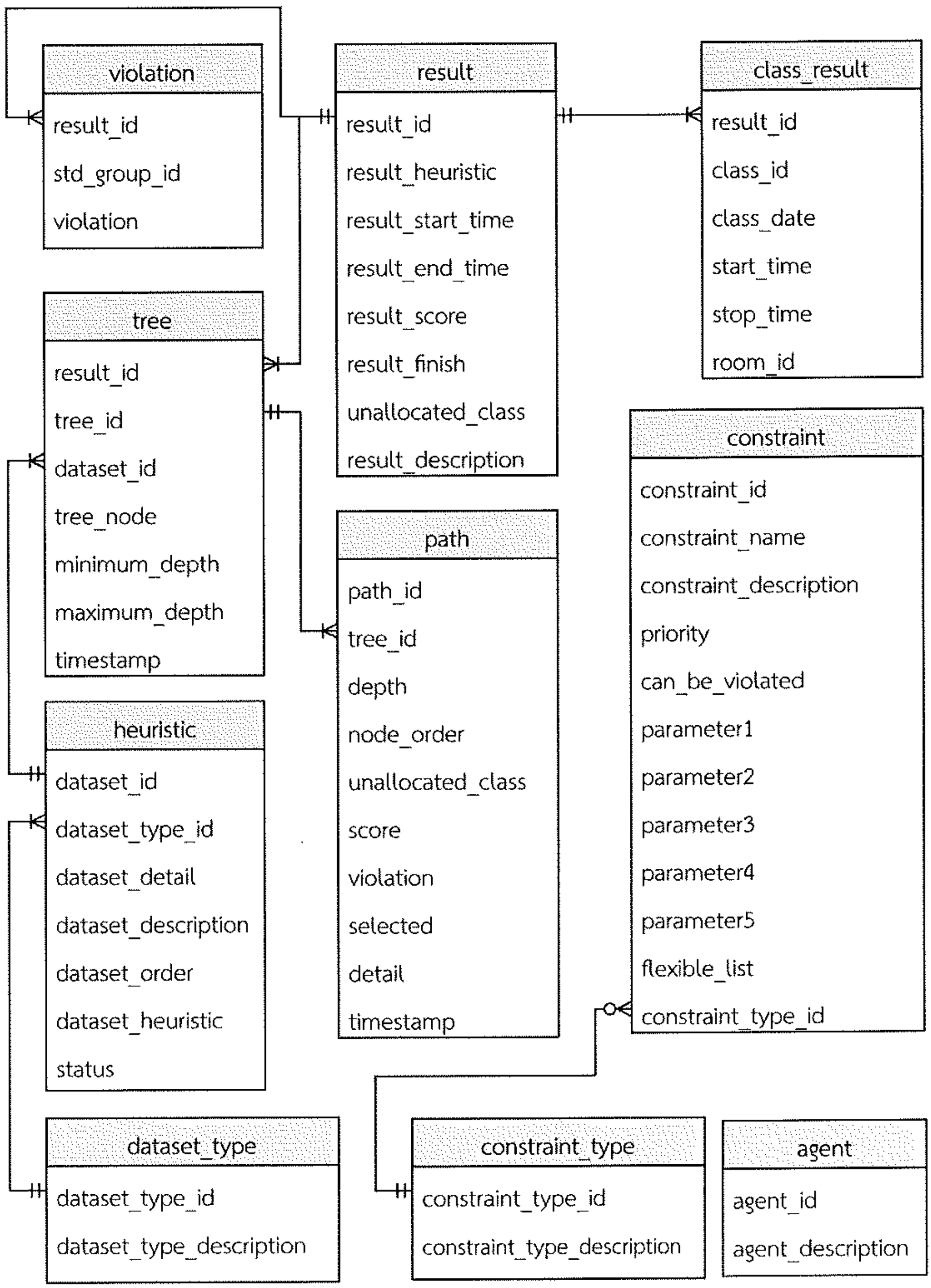


Figure B-2 The database structure for the searching data

2) tree: This table stores the search tree information from the search engine. The table fields are described follows.
   - result_id is the index of result table in an integer format.
   - tree_id is the index of tree table in an integer format.
   - dataset_id is the index of the dataset used to create the search tree from heuristic table in an integer format.
   - tree_node is a list of classes used to create the search tree in a text format.
   - minimum_depth is the minimum depth to discrepancy the heuristics in an integer format.
   - maximum_depth is the maximum depth to discrepancy the heuristics in an integer format.
   - timestamp is the timestamp of each record in a timestamp format.
3) path: This table stores information of each path of the search tree. The table fields are described follows.
   - path_id is the index of path table in an integer format.
   - tree_id is the index of tree table in an integer format.
   - depth is the depth of DDS when searching through the path in an integer format.
   - node_order is the order of nodes in the path in a text format.
   - unallocated_class is the path unallocated classes list in a text format.
   - score is the score of the path in a text format.
   - violation is the violation list in a text format, The example pattern of the text is "constraint_name:number,constraint_name:number"
   - selected is the status of the path in an integer format. The value can be 0 or 1 represents unselected and selected, respectively.
   - detail is the log of DDS for recording some remarks in a text format.
   - timestamp is the timestamp of each record in a timestamp format.
4) violation: This table stores the number of constraint violations from the result each student group. The table fields are described follows.
   - result_id is the index of result table in an integer format.

- std_group_id is the index of student_group table in an integer format.
- violation is the number of constraint violations from the result in a text format. The example pattern of the text is "constraint_id:number, constraint_id:number".

5) heuristic: This table stores the datasets for creating a search tree and define the order of dataset on the searching process. The table fields are described follows.
   - dataset_id is the index of heuristic table in an integer format.
   - dataset_type_id is the index of dataset_type table in an integer format.
   - dataset_detail is the dataset list in a text format. For example, if the heuristic contains two teachers then the text will be "teacher_id, teacher_id".
   - dataset_description is the dataset description in a text format.
   - dataset_order is the order of the dataset when searching in an integer format.
   - dataset_heuristic is the ordering heuristic of the dataset in an integer format.
   - status is the status of the dataset in an integer format. The value can be 0 or 1 represents unavailable and available, respectively.
6) dataset_type: This table stores the dataset type. The table fields are described follows.
   - dataset_type_id is the index of dataset_type table in an integer format.
   - dataset_type_description is the description of the dataset type in a text format
7) constraint: This table stores the constraints defined by the user. The table fields are described follows.
   - constraint_id is the index of constraint table in an integer format.
   - constraint_name is the constraint code in a text format.
   - constraint_description is the title of the constraint in a text format.
   - priority is the priority of the constraint in an integer format.

- can_be_violated is the property of the constraint in an integer format. The value can be 0 or 1 represents hard and soft constraint, respectively.
- parameter1 is the scope of the constraint in a text format. The value is agent_id from agent table.
- parameter2 is list of the agents in the scope in a text format. For example, if the scope of the constraint is two room type then the text will be "room_type_id,room_type_id".
- parameter3 is depended on the constraint type. The value is stored in a text format.
- parameter4 is depended on the constraint type. The value is stored in a text format.
- parameter5 is depended on the constraint type. The value is stored in a text format.
- flexible_list is the flexible list in a text format. The list contains classes which can be violated even it is the hard constraint. For example, if the flexible contains three classes then the text will be "class_id,class_id, class_id".
- constraint_type_id is the index of the constraint_type table.

8) constraint_type: This table stores all constraint types which can be defined to the search engine. The table fields are described follows.
    - constraint_type_id is the index of constraint_type table in an integer format.
    - constraint_type_description is the constraint type description words in a text format.
9) agent: This table stores all agent types which can be defined as the scope of the constraints. The table fields are described follows.
    - agent_id is the index of agent table in an integer format.
    - agent_description is the agent type name in a text format.

APPENDIX C
THE PROTOTYPE CONFIGURATIONS AND RESULTS

The configuration and the result pages of the proposed prototype are shown in this appendix include the constraint configuration, the heuristic configuration and the result pages.

C-1. The constraint configuration pages

To define the constraints of the prototype, the user should start with the homepage of the configuration website. The hyperlink to access the constraint configuration page is placed at the A sign area in Figure C-1.

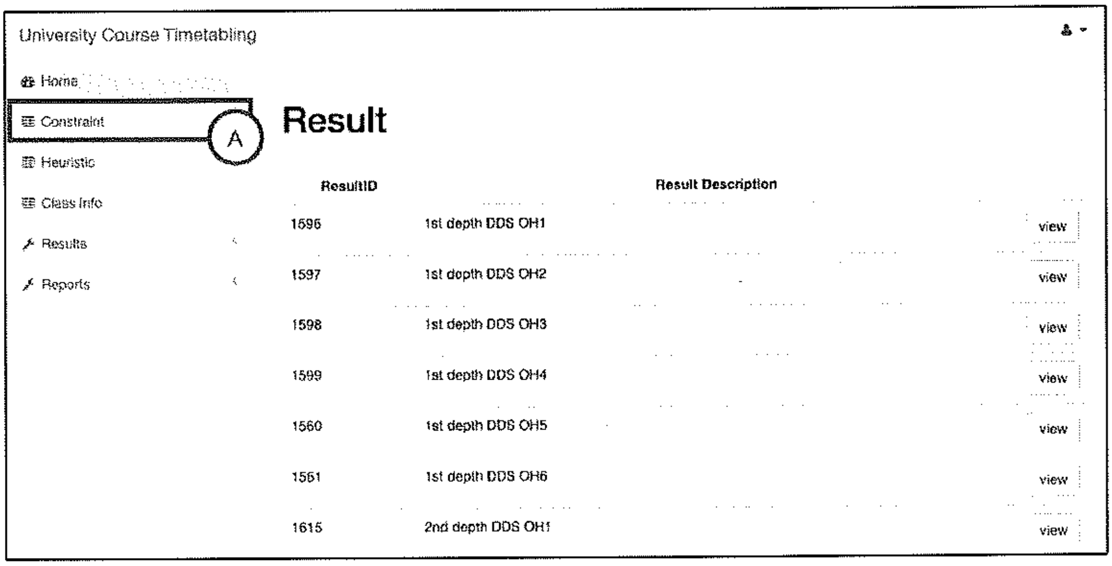


Figure C-1 The homepage view

Figure C-2 shows the components in the constraint configuration page. The constraints are listed inside the panel with the code and description. The A sign is a toggle button to enable or disable the constraint. The button at the B sign is an expanded button to show the scope and the flexible list of the constraint. The expanded constraint view is shown in Figure C-3. The A sign area shows a scope of the constraint, while the B sign area shows the flexible list of the constraints

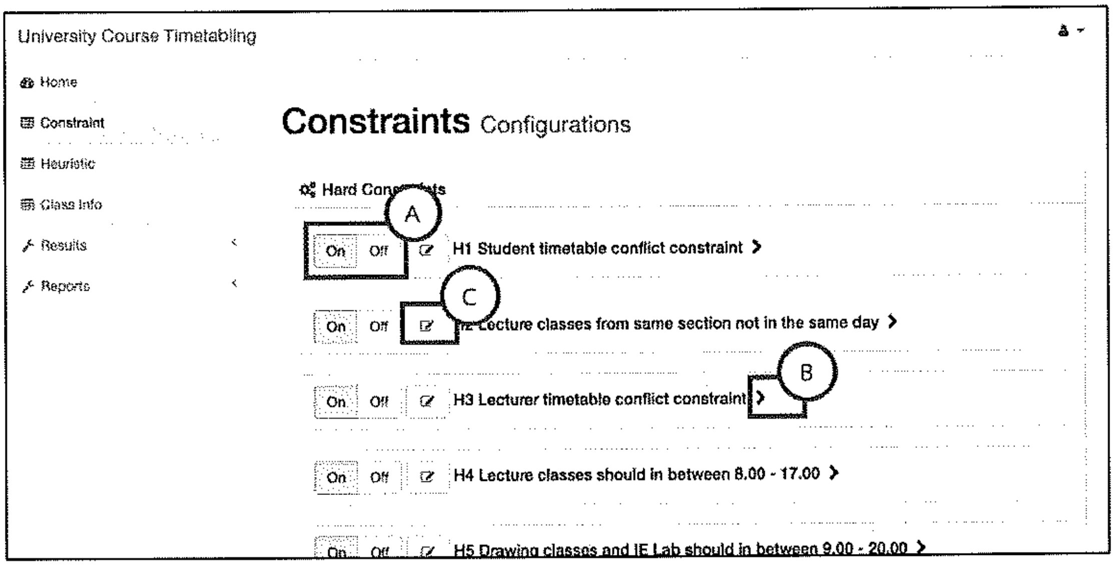


Figure C-2 Constraint list page

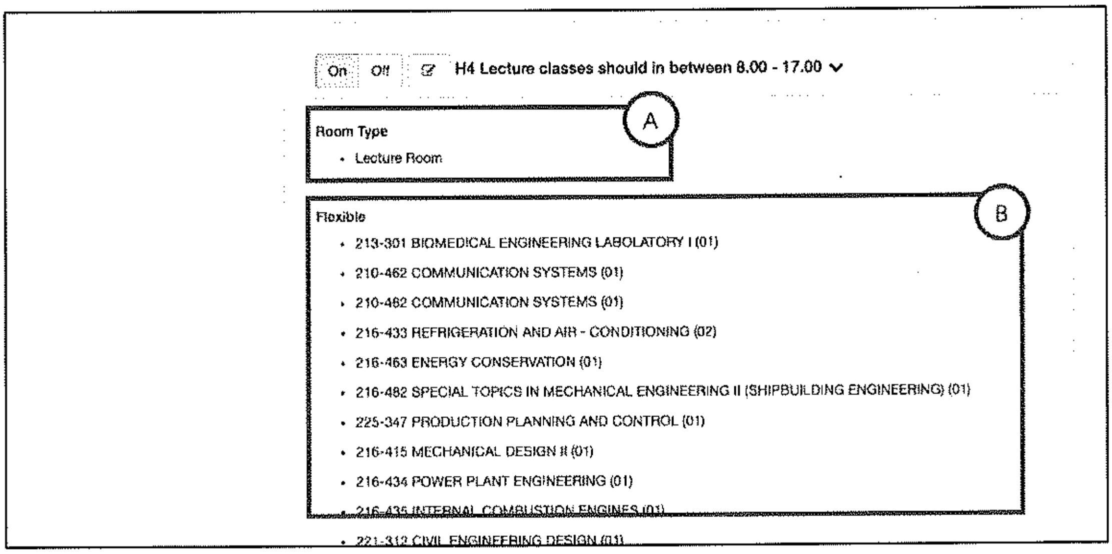


Figure C-3 Expanded constraint view in the constraint list

The button at the C sign in Figure C-2 is a link to the constraint detail page which is shown in Figure C-4. The user can change the attributes of the constraint by this page. The A sign in Figure C-4 is constraint type selector. The prototype supports eight types of constraints which are described in section 3.6. The time constraint type is used as an example. The B sign textbox is the constraint code which is a short name of the constraint. The text area at the C sign is used for a full description of the constraint. The D sign toggle button is used for defining the hard/soft constraint. The priority of the constraint can be defined in the E sign textbox.

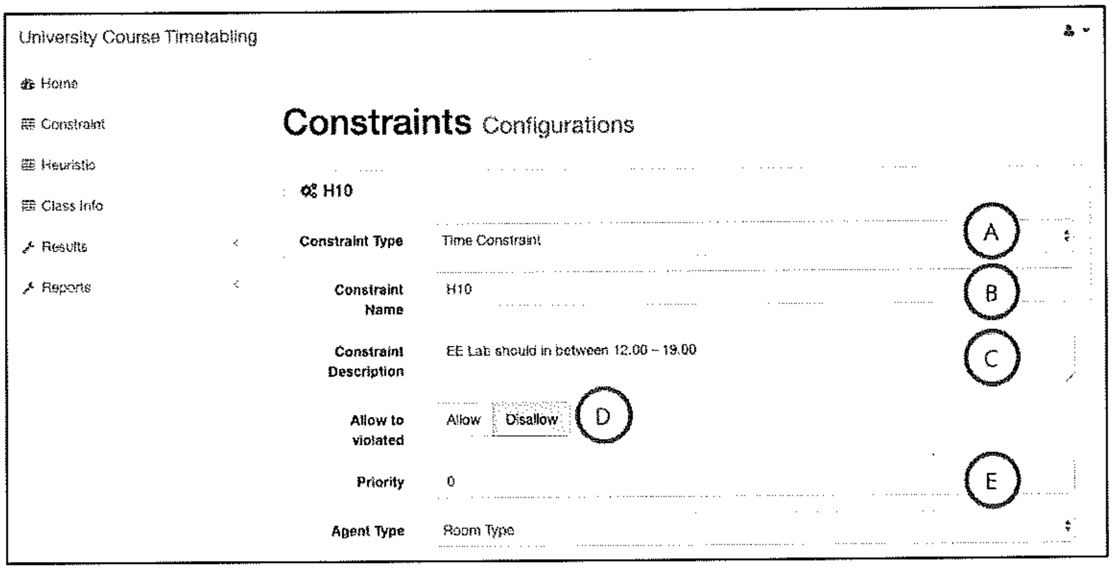


Figure C-4 Constraint configuration page

Figure C-5 shows the configuration of the constraint scope. The scope can be defined in eight types include class, section, class type, room type, teacher, student group and class size. The B sign textbox allows the user to input some searching phrase before pressing the C sign button for searching the agent to define a scope of the constraint which is shown in D sign area. After press the C sign button, the agent list will be shown in the E sign area. The user can check the checkbox to include the agent to the scope or uncheck to remove the agent from the scope. The F sign toggle button is specific for the time constraint used to add a restriction to the timeslots. The timeslot can be selected in the G sign area.

The user can also include the flexible list for a class which is allowed to have the hard constraint violation with the component in Figure C-6. The A sign text box is used as an input for searching phrase. The user can search for including the class to the flexible list. The flexible list is shown in the C sign area while the user can change the classes in the flexible list by the checkboxes in the D sign area. After changing the attributes of constraint, the user can save the configuration by pressing the E sign button.

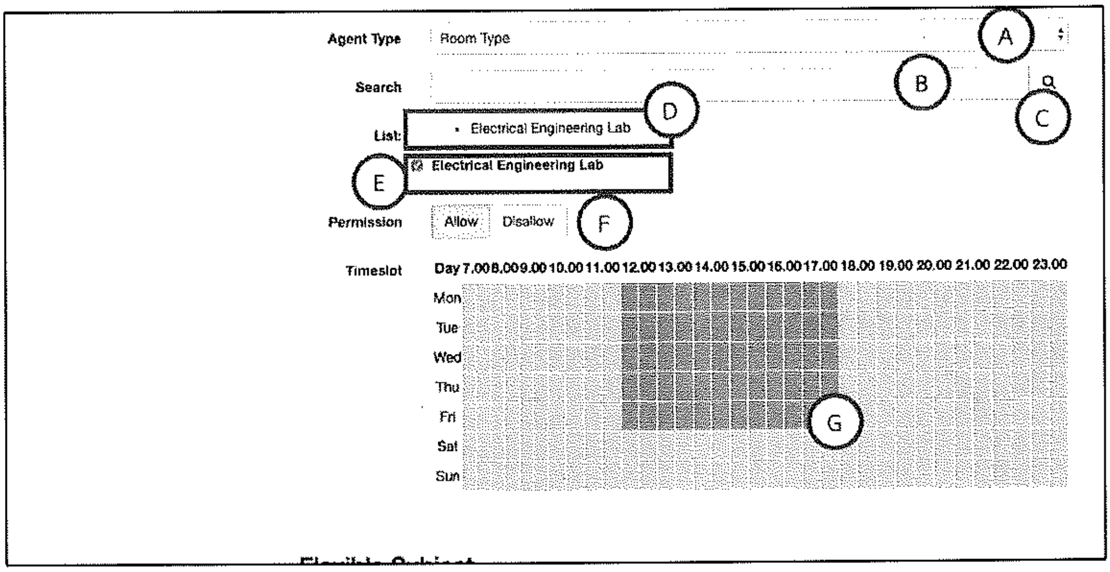


Figure C-5 Specific scope of the constraint

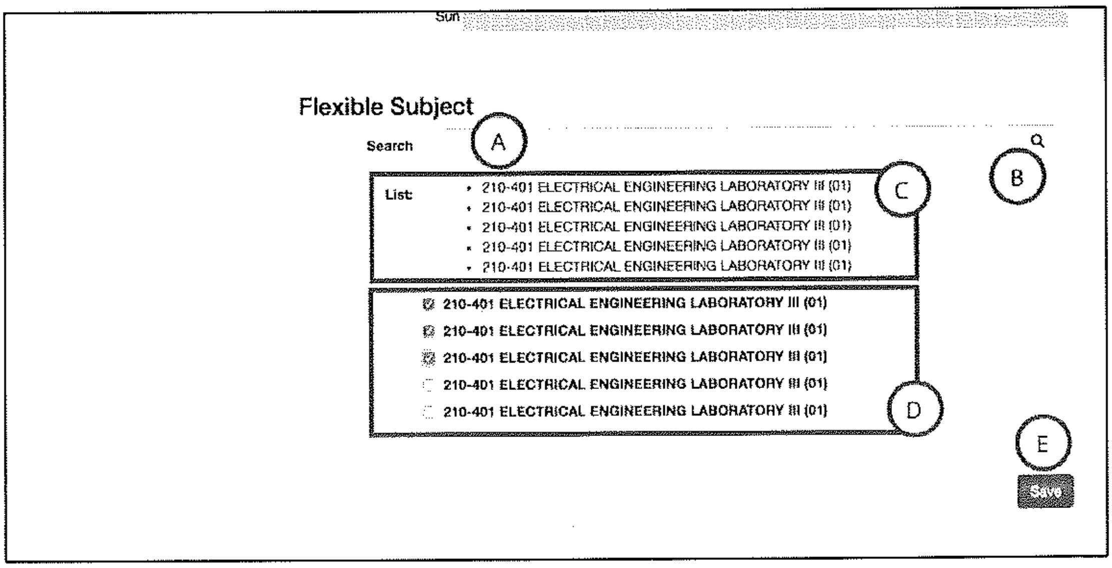


Figure C-6 Edit the flexible list

C-2. The heuristic configuration pages

The hyperlink to the heuristic configuration is on the sidebar at A sign area in Figure C-7. The description of each dataset is shown in B sign area. The user can change the order of the dataset by the up/down button at the C sign area. The user can add new dataset by pressing the E sign button and change the attributes of the dataset by the D sign button. The dataset configure page is shown in Figure C-8.

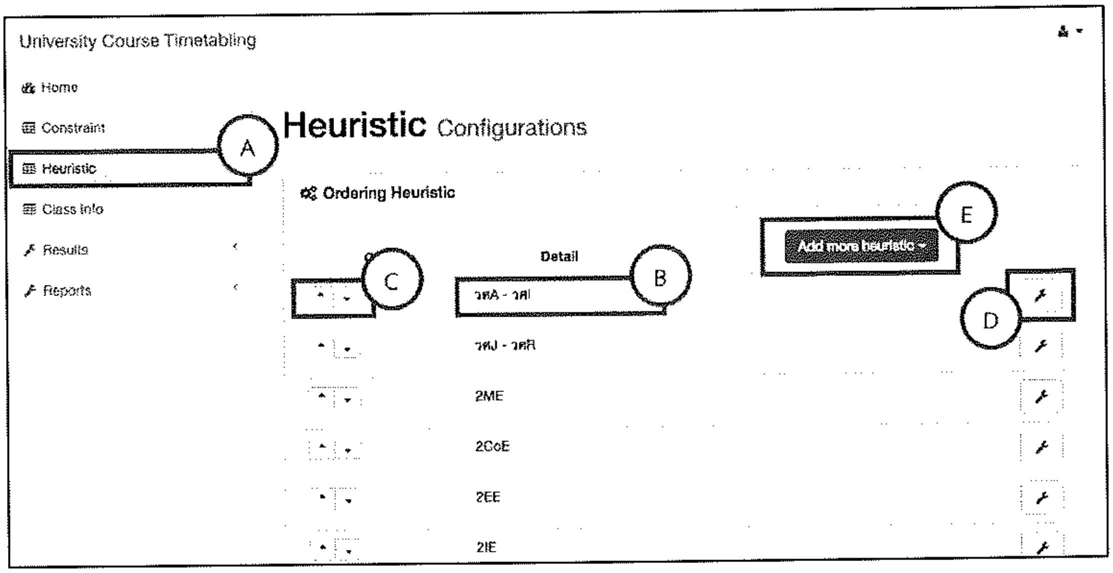


Figure C-7 Heuristic list page

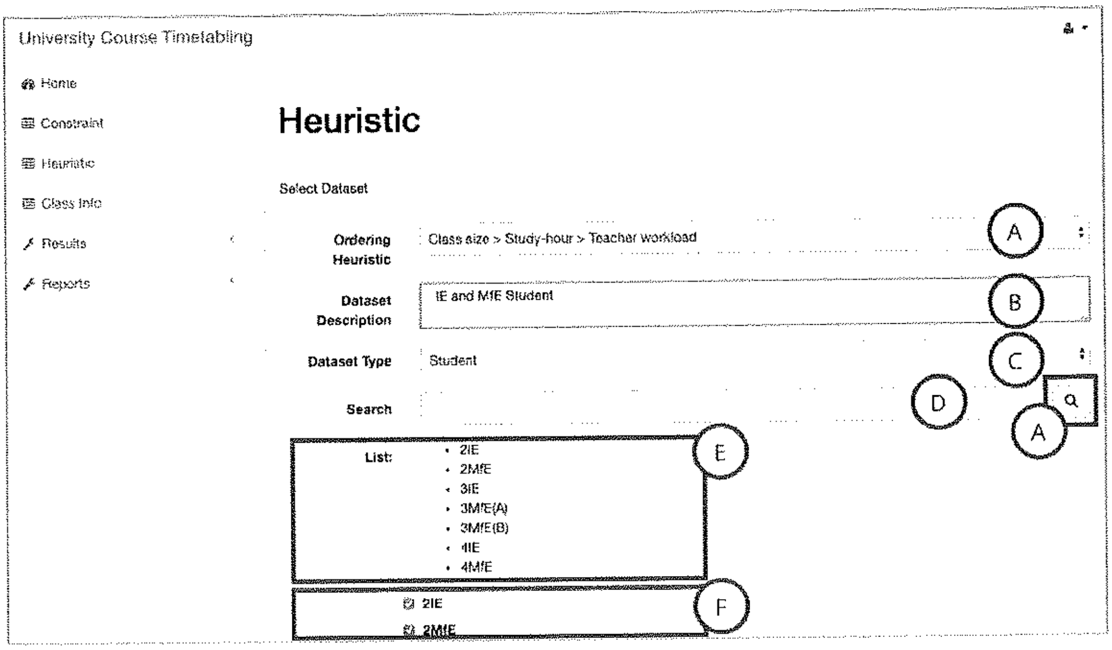


Figure C-8 Insert or edit the dataset in the heuristic

The user can select the ordering heuristic of the dataset at the A sign selector in Figure C-8. The user can add the description for the dataset in the B sign text area. The C sign selector is used to select the dataset type. After selecting the dataset type, the user can search for the agent to insert into the dataset. The search result and the current agents in the dataset are shown as a checkbox in F sign area. The user can add the agent to the dataset by checking the checkbox or remove the agent by unchecking the checkbox. The current dataset agents are listed in the E sign area.

C-3. The result pages

The result pages are described in this section. The results contain three categories of timetable include student timetable, teacher timetable and room timetable.

The student result page is shown in Figure C-9 and Figure C-10. The Figure C-9 shows the start page of student timetable result. To access the student result page, the user has to expand the results panel by the A sign button on the left sidebar. After that, the student timetable hyperlink will be shown. The C sign textbox is used for adding the code of the result. The student group name can be inserted to the D sign textbox. Then, the user can press the E sign button to get the student timetable from the result.

Figure C-9 shows the electrical engineering student timetable from the result code 1615. The B sign area shows the student group name of the timetable. The C and D sign area show timeslot and date of the timetable, respectively. The pre-define class is shown in a lighter color as the E sign area. While, the result from the prototype is shown in darker color as the F sign area. The user can change the student group timetable or the result by change the student group and result code in the A sign and G sign textbox before pressing the H sign button.

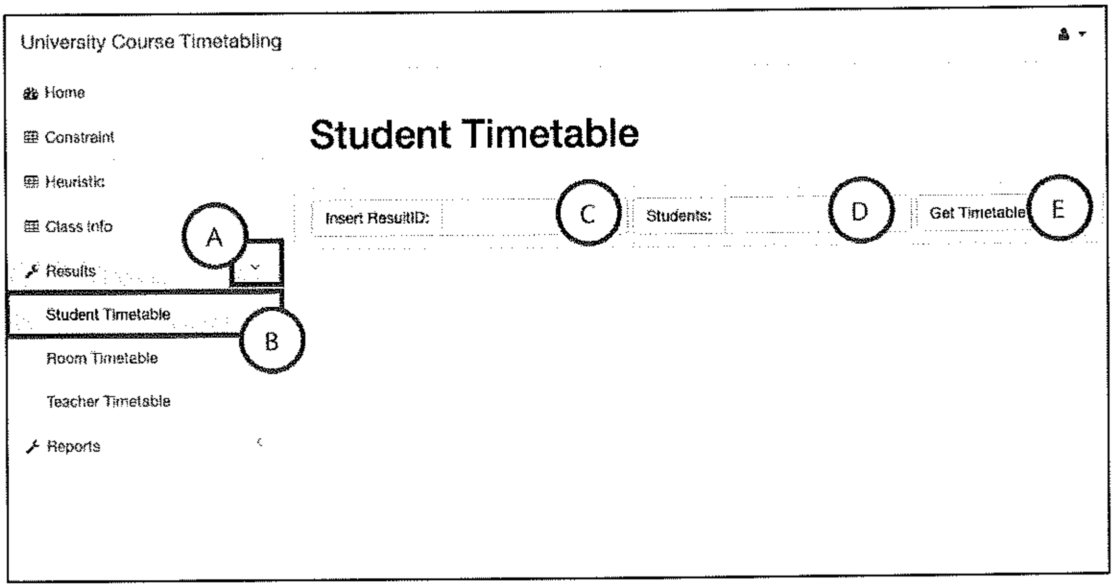


Figure C-9 Search student timetable from the results

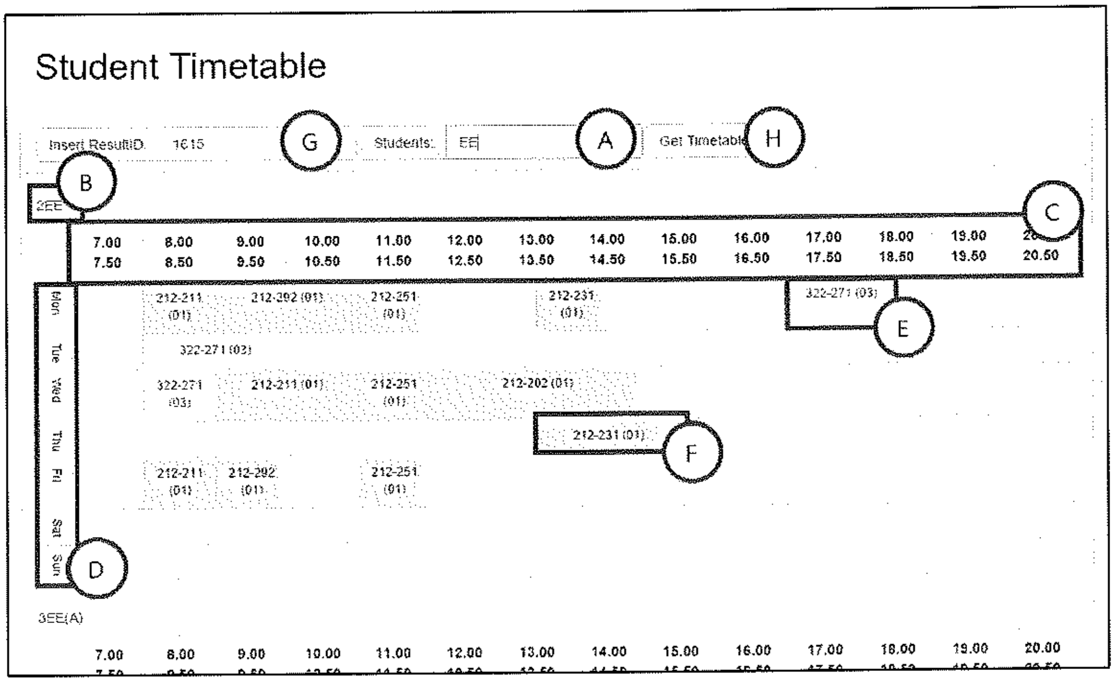


Figure C-10 Student timetable view

The teacher result page is shown in Figure C-11 and Figure C-12. The Figure C-11 shows the start page of teacher timetable result. To access the teacher result page, the user has to expand the results panel by the A sign button on the left sidebar. After that, the teacher timetable hyperlink will be shown in the third order of the appearing list at the B sign area. The C sign textbox is used for adding the code of the result. The teacher name can be inserted to the D sign textbox. Then, the user can press the E sign button to get the teacher timetable from the result.

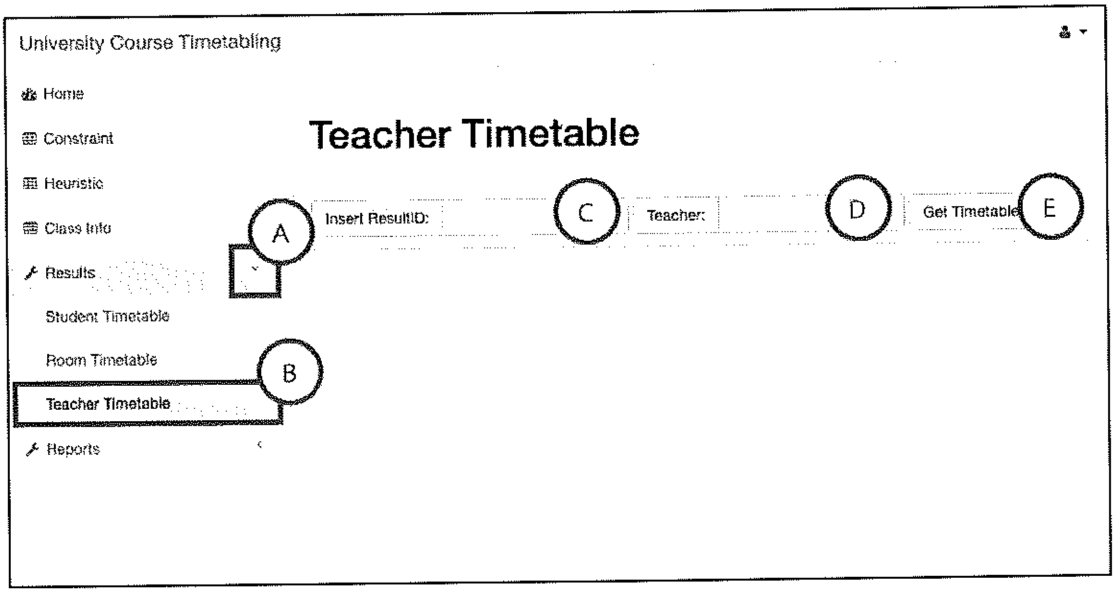


Figure C-11 Search teacher timetable from the results

Figure C-12 shows the timetable of Vitaya Mhadnui from the result code 1615. The B sign area shows the teacher name. The timetable structure is as same as the student timetable, which is described in Figure C-10. The user can change the teacher timetable or the result by change the teacher name and result code in the A sign and D sign textbox before pressing the C sign button.

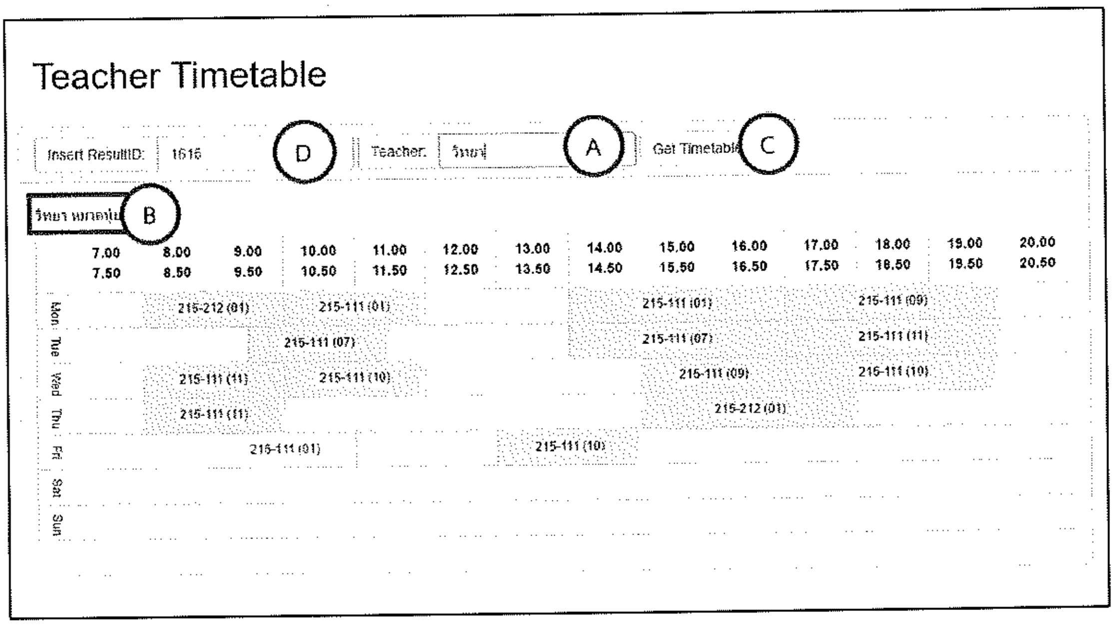


Figure C-12 Teacher timetable view

The room result page is shown in Figure C-13 and Figure C-14. The Figure C-13 shows the start page of room timetable result. To access the room result page, the user has to expand the results panel by the A sign button on the left sidebar. After that, the room timetable hyperlink will be shown at the B sign area. The C sign textbox is used for adding the code of the result. The room name can be inserted to the D sign textbox. Then, the user can press the E sign button to get the room timetable from the result.

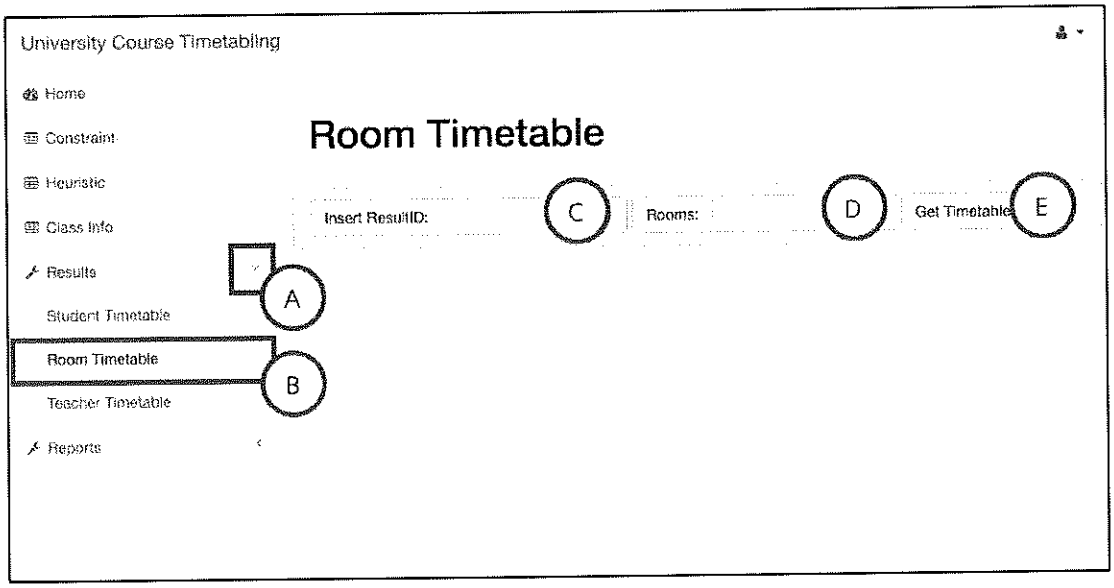


Figure C-13 Search room timetable from the results

Figure C-14 shows the A200 timetable from the result code 1615. The B sign area shows the room name beside with the room index inside the bracket. The timetable structure is as same as the student timetable which is described in Figure C-10. The user can change the room timetable or the result by change the room name and result code in the A sign and C sign textbox before pressing the D sign button.

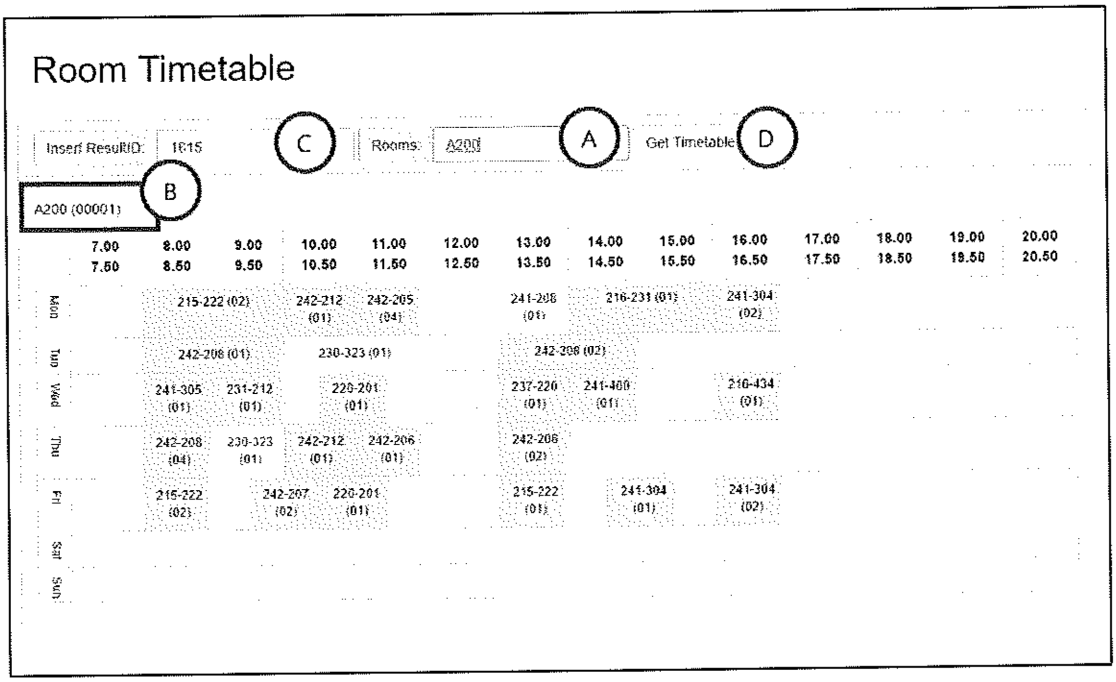


Figure C-14 Room timetable view

VITAE

Name Warakorn Sitthirit

Student ID 5410120106

Educational Attainment

Degree	Name of Institution	Year of Graduation
B.Eng. (Computer Engineering)	Prince of Songkla University	2010

Scholarship Awards during Enrolment

Prince of Songkla University, Funding no. ENG550405S-0 and PRPM ID 12034

List of Publication and Proceeding

W. Sitthirit, S. Vasupongayya, "Applying a mixed objective model in a university timetabling solution searching technique", in Proceedings of *2013 International Computer Science and Engineering Conference (ICSEC)*, IEEE, 2013