โครงการ

การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ ปัญหาการจัดตารางห้องเรียนด้วยกำหนดการเชิงเส้นจำนวนเต็ม
Classroom scheduling problem with integer linear program

ชื่อนิสิต นางสาวชนิดา ลีลายุทธ์โท **เลขประจำตัว** 5933511023

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์
สาขาวิชา คณิตศาสตร์

ปีการศึกษา 2562

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปัญหาการจัดตารางห้องเรียนด้วยกำหนดการเชิงเส้นจำนวนเต็ม

นางสาวชนิดา ลีลายุทธ์โท เลขประจำตัว 5933511023

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

Classroom scheduling problem with integer linear program

Miss Chanida Leelayutto

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science Program in Mathematics
Department of Mathematics and Computer Science
Faculty of Science Chulalongkorn University
Academic Year 2019

หัวข้อโครงงาน	ปัญหาการจัดตารางห้องเรียนด้วยกำหนดการเชิงเส้นจำนวนเต็ม
โดย	นางสาวชนิดา ลีลายุทธ์โท
สาขาวิชา	คณิตศาสตร์
อาจารย์ที่ปรึกษาโครงงานหลัก	รองศาสตราจารย์ ดร.พันทิพา ทิพย์วิวัฒน์พจนา

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับโครงงานฉบับนี้เป็นส่วนหนึ่ง ของการศึกษาตามหลักสูตรปริญญาบัณฑิต ในรายวิชา 2301499 โครงงานวิทยาศาสตร์ (Senior Project)

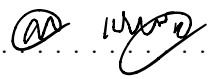
. . . . . หัวหน้าภาควิชาคณิตศาสตร์
และวิทยาการคอมพิวเตอร์

(ศาสตราจารย์ ดร.กฤษณะ เนียมมณี)

คณะกรรมการสอบโครงงาน

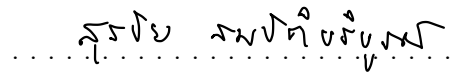
. . . . . อาจารย์ที่ปรึกษาโครงงานหลัก

(รองศาสตราจารย์ ดร.พันทิพา ทิพย์วิวัฒน์พจนา)



. . . . . กรรมการ

(รองศาสตราจารย์ ดร.ศจี เพียรสกุล)



. . . . . กรรมการ

(ผู้ช่วยศาสตราจารย์ สุรชัย สมบัติบริบูรณ์)

ชนิดา ลีลายุทธ์โท: ปัญหาการจัดตารางห้องเรียนด้วยกำหนดการเชิงเส้นจำนวนเต็ม. (CLASSROOM SCHEDULING PROBLEM WITH INTEGER LINEAR PROGRAM)
อ.ที่ปรึกษาโครงงานหลัก: รองศาสตราจารย์ ดร.พันทิพา ทิพย์วิวัฒน์พจนา, 71 หน้า.

ในโครงงานนี้เราได้นำกำหนดการเชิงเส้นจำนวนเต็มเข้ามาแก้ปัญหาการจัดตารางห้องเรียน โดยเราได้กำหนดค่าความพึงพอใจของช่วงเวลาที่ต้องการให้เกิดการเรียนการสอนของแต่ละวิชาไว้ ซึ่งในการจัดตารางห้องเรียนครั้งนี้ เราคาดหวังผลลัพธ์ที่สอดคล้องกับค่าความพึงพอใจดังกล่าวมากที่สุดและต้องเป็นไปตามเงื่อนไขที่ทำให้ตารางนี้มีประสิทธิภาพ เช่น ไม่ควรกำหนดวิชาเรียนในห้องที่ไม่พร้อมใช้งาน อาจารย์ไม่สามารถสอนหลายวิชาในเวลาเดียวกันได้ และห้ามนิสิตเรียนหลายวิชาในเวลาเดียวกัน เป็นต้น โดยเราใช้ข้อมูลของภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในภาคการศึกษาต้น ปีการศึกษา 2562 ซึ่งข้อมูลดังกล่าวได้มีการระบุผู้สอนสำหรับแต่ละวิชาไว้แล้ว นอกจากนั้นยังมีข้อมูลเวลาการใช้งานได้ของห้องเรียนแต่ละห้อง จากนั้นเราได้ทำการจัดเรียงข้อมูลใหม่เพื่อให้ง่ายต่อการดึงไปแก้ปัญหาใน CPLEX Studio IDE software version 12.6.3 และ Python version 3.7 ทำให้ได้การจัดตารางห้องเรียนตามเงื่อนไขและได้ผลลัพธ์ที่ตอบสนองความพึงพอใจมากที่สุด

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต . . . . . Chanida L. . . . . .
สาขาวิชา . คณิตศาสตร์ . ลายมือชื่อ อ.ที่ปรึกษาโครงงานหลัก . . . . . .
ปีการศึกษา . . . . 2562 . . . .

5933511023 : MAJOR MATHEMATICS. CHANIDA LEELAYUTTO: CLASSROOM SCHEDULING PROBLEM WITH INTEGER LINEAR PROGRAM.
ADVISOR: ASST PROF.PHANTIPA THIPWIWATPOTJANA, Ph.D., 71 pp.

In this project, we use an integer linear program to solve the classroom scheduling problem. We set the preference of time slots for each course. In this classroom scheduling, we expect that our result would be corresponding to the setup preference. Moreover, the result must follow some restrictions to make an efficient schedule such as class meetings should not be assigned to busy rooms and each lecturer and each student must not have more than one class meeting at a time. We use the data of first-semester in 2019 of the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University as a case study. The data consist of courses with specified lecturers and available time slots of each room. Then we adjust data to make them easier for solving the problem in CPLEX Studio IDE software version 12.6.3 and Python version 3.7, which provide the classroom schedule under the constraints and the objective of maximizing overall preference.

Department . Mathematics and Computer Science . Student's Signature Chanida L.
Field of Study . . Mathematics . . Advisor's Signature P. T.
Academic Year . . . . . 2019 . . . . .

Acknowledgements

This project could not have been achieved without help from the following important persons.

Firstly, I would like to express my sincere thanks to my advisor, Associate Professor Phantipa Thipwiwatpotjana, Ph.D. for her invaluable help, advice, encouragement throughout the project.

Secondly, I would like to express my committee Associate Professor Sajee Pianskool, Ph.D. and Assistant Professor Surachai Sombatboriboon for giving the comment and advice.

Finally, I most gratefully acknowledge my family, my teachers, and my friends for all their suggestions and support throughout the project.

Chanida Leelayutto

Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Background and Rationale

A scheduling problem is a problem that arises in a wide variety of fields, such as a schedule of medical personnel, a schedule of part-time staff, and a schedule of public transport. If we have a good method or strategy to manage conflicts of a particular field, we should be able to create a reasonable schedule for that field. This project is interested in classroom scheduling which is commonly encountered by many universities.

Classroom scheduling problem is very difficult to solve by hand. People normally replicate the timetables of the previous years with some changes to create a new timetable in the same organization but it may be not quite efficient. Classroom scheduling has an important role in the development of education both budget and efficiency. For example, if the capacity of the assigned classroom is less than or much larger than the number of students, it will cause no seats or too many seats for students which will lead to budget or efficiency problems.

This project tries to handle the restrictions such as the limited number of classrooms, the available lecturers/ classrooms at each time period, and number of class meetings assigned to each lecturer/ student at each time slot. Then, we apply an integer linear program learned in the course of Operations Research II to find an optimal solution to this problem.

1.2 Objectives

To solve the classroom scheduling problem by using CPLEX Studio IDE software version 12.6.3 and Python version 3.7.

1.3 Scopes

1. We use the data from the first semester of 2019 provided by the Office of the registrar and the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University.
2. We consider only courses in the Department of Mathematics and Computer Science and classrooms in Maha Vajirunhis, Mahamakut, and Tab Nilaniti buildings.
3. We consider only courses that are taught between 8 a.m. - 5 p.m during the weekdays.
4. We use CPLEX Studio IDE software version 12.6.3 and Python version 3.7 to solve our classroom scheduling problem.

1.4 Project Activities

1. Study classroom scheduling problem from literature review.
2. Combine related data.
3. Filter constraints from literature review that are reasonable for the Department of Mathematics and Computer Science.
4. Build an integer linear programming model for solving the problem.
5. Study how to use CPLEX Studio IDE software version 12.6.3.
6. Code the model on CPLEX Studio IDE software version 12.6.3 and Python 3.7.

7. Recheck and modify the model.

8. Conclude the results and write a report.

1.5 Benefits

1. The benefits of the project owner.

   (a) Be able to apply knowledge from the Operations Research II course in the classroom scheduling problem.

   (b) Be able to use CPLEX Studio IDE software version 12.6.3 and Python 3.7 to solve the classroom scheduling problem.

2. The benefits of project users.

   (a) Increase convenience and efficiency of using classroom schedule as a result of this project.

   (b) Can apply the integer linear program with different data to solve the classroom scheduling problem.

This project is divided into 5 chapters as follows. Chapter 2 explains background knowledge and literature review. Then, Chapter 3 mentions data from the department, and how to adjust the data for our model and describe the model. Next, Chapter 4 explains CPLEX program, Python program, and our result of the classroom scheduling problem. Finally, the last chapter serves for conclusion and suggestions.

Chapter 2
Background knowledge and Literature review

In this chapter, we present related knowledge of our project including integer linear program and literature review on listed constraints of a classroom scheduling problem in each research paper.

2.1 Integer linear program

An integer linear program is a mathematical model consisting of linear restrictions and linear objective function with integer variables. We can write an integer linear programming problem in the following form:

$$\begin{aligned} \max / \min \quad & c^T x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \\ & x \in \mathbb{N} \cup \{0\}, \end{aligned}$$

where x is a vector of decision variables,
c is a vector objectives function coefficients,
b is a vector of right hand side values,
and A is a matrix coefficient of decision variables.

Normally, people solve an integer linear program by many methods. There are two well known methods of getting an exact optimal solution: cutting plane and branch and bound methods. Cutting plane method acts by cutting the feasible region of the linear programming relaxation, then optimal integer solution becomes an extreme point and can be found by the simplex method. Branch and bound method

will branch a fractional solution of $x_i = \frac{a}{b}$ with $0 < a < b$ into two new relaxation programs, each with additional constraint of $x_i \leq \lfloor \frac{a}{b} \rfloor$ or $x_i \geq \lceil \frac{a}{b} \rceil$, then use an integer solution of a relaxed problem as a boundary of the optimal objective value of the original problem. More details on integer linear program, cutting plane, and branch and bound methods could be seen in [3], [9], and [8].

2.2 Literature review

Classroom scheduling problem is an important problem in the education field. So many people intend to solve this problem. As a result, this problem is solved by many methods and may have different restrictions. There are many research papers with different methods and constraints for the problem. Here are some reviews on the methods have been done so far on the topic of classroom scheduling problem.

In 2004, S. Daskalaki, T. Birbas, and E. Housos [11] used integer programming to solve the problem where binary decision variables consist of courses, students, teachers, days and time slots. In 2005, P. Kostuch [7] solved the problem by meta-heuristics that is divided into 3 phases: feasibility, slot sequence, and exchange of events. Then, P. Avella and I. Vasil'EV [6] used a branch-and-cut algorithm to solve the problem.

In 2007, A. Gunawan, K.M. Ng, and K.L. Poh [1] presented a hybrid algorithm for solving a timetabling problem that combines both teacher assignment and course scheduling problems simultaneously. Next, S. Ghaemi, M.T. Vakili, and A. Aghagolzadeh [12] solved the problem by the evolutionary computation algorithms, a genetic algorithm that tries to minimize the number of conflicts in the timetable. They assigned courses to a time slot that satisfy given constraints. Then, S. Abdullah, E.K. Burke, and B. McCollum [10] presented evolutionary-based approaches with local search to solve the problem.

In 2011, A. Oner, S. Ozcan, and D. Dengi [2] used a hybrid algorithm consisting of a heuristic graph node coloring and artificial bee colony to solve the problem. In 2014, N.A.H. Aizam and L. Caccetta [5] constructed a general model for university

course timetabling. Their model combines 17 different constraints from other literature and adds 3 new constraints that should be part of the restrictions in a general university-based timetabling model. In 2018, M. Assi, B. Halawi, and R.A. Haraty [4] used the genetic algorithm and graph coloring to solve the problem.

This project focuses on the integer linear programming by selecting constraints that are reasonable for the Department of Mathematics and Computer Science, Chulalongkorn University. We apply the model from research papers to build our model in CPLEX. At this end, we summarize all constraints used in the above listed research papers in Table 2.1 as our reference. We add the remark in the last column of Table 2.1 that we use that particular constraint as the constraint in our project.

Table 2.1: The constraints from research papers

Constraint	[1]	[2]	[4]	[5]	[6]	[7]	[10]	[11]	[12]	Our model
Lecturer										
1. All the class meetings taught by a lecturer will be spread throughout a week.	✓									
2. Each class meeting can be taught by one lecturer.	✓									
3. Each lecturer cannot be assigned to more than one class meeting in the same time slot.	✓	✓	✓	✓	✓			✓	✓	CPLEX(2)
4. Each lecturer should be assigned to so many teaching periods as his/her weekly teaching load requires.								✓		
5. Each lecturer teaches at least one course per semester and cannot teach more than the number of current courses.	✓									Excel
6. Each lecturer will not be assigned courses that he/she unable to teach.	✓									
7. Maximum consecutive class meetings for a lecturer.				✓						CPLEX(10)
8. Maximum total class meetings that a lecturer can teach in a day.			✓	✓	✓					CPLEX(8)
9. The lecturers should have a day off.		✓								

Constraint	[1]	[2]	[4]	[5]	[6]	[7]	[10]	[11]	[12]	Our model
Lecturer										
10. The class meetings should not be allocated to a time period inconvenient for a lecturer.				✓	✓				✓	CPLEX(4)
11. The number of class meeting hours assigned for each course should be the same as the weekly number of class meeting hours stated in the curriculum.		✓								
12. The number of lecturers who can teach each course is limited.	✓									
13. The preferences of the lecturers about the assigned time slots should be taken into consideration.		✓			✓				✓	
Student										
14. A student has a class meeting in the last slot of the day.						✓	✓			
15. A student has more than two consecutive courses.						✓	✓			
16. Each student group cannot be assigned to more than one class meeting in the same time slot.		✓	✓	✓	✓	✓	✓	✓	✓	CPLEX(3)
17. Each student study more one course in a day.			✓			✓	✓			
18. Maximum consecutive class meetings for a student.				✓						CPLEX(11)

Constraint	[1]	[2]	[4]	[5]	[6]	[7]	[10]	[11]	[12]	Our model
Student										
19. Maximum total class meetings that a student can attend in a day.				✓	✓					CPLEX(9)
20. The empty periods between any two class meetings of each student are not allowed.					✓					
21. The number of students in each class meeting does not exceed the capacity of the classroom.		✓	✓			✓	✓			Python
22. The preferences of the students about the assigned time slots should be taken into consideration.		✓								
Room										
23. All the hours of a class meeting scheduled in a day should be located in the same room.					✓					Python
24. Only one class meeting can be assigned to a classroom in a time slot.		✓	✓		✓	✓	✓	✓	✓	Python
25. The room should satisfy the features required by the course.		✓			✓	✓	✓			Excel

Constraint	[1]	[2]	[4]	[5]	[6]	[7]	[10]	[11]	[12]	Our model
Room										
26. The number of class meetings taught cannot exceed the number of classrooms available during each time period.	✓			✓	✓					CPLEX(5)
Class meeting										
27. All class meetings have to be spread throughout a week.	✓			✓						
28. All class meetings must be assigned to a time slot.	✓			✓	✓			✓		CPLEX(1)
29. Class meetings need to occur consecutively.				✓						
30. Class meetings should not occur consecutively.				✓						
31. Each course should be assigned to a given period in a given day.								✓		
32. Each course should be scheduled for as many teaching periods as the curriculum of each group of students requires.								✓		
33. For each course, only one section can be conducted in every time period.	✓									
34. For each student, the major and minor required courses should be scheduled in non-overlapping slots.			✓							
35. Having a day off between two class meetings.				✓					✓	CPLEX(14)

Constraint	[1]	[2]	[4]	[5]	[6]	[7]	[10]	[11]	[12]	Our model
Class meeting										
36. If a class meeting is scheduled in a day, it should take between minimum and maximum daily number of teching hours.					✓			✓		
37. No course should be allocated to a time period that heads of department don't demand because of other work.									✓	
38. Some specific class meetings must be scheduled after another class meeting.				✓						
39. Some specific class meetings must be scheduled in the morning or afternoon in the same day.				✓	✓					
40. Some specific class meetings must be scheduled in the same day.				✓						
41. Some specific class meetings must be scheduled in the same time slot.				✓						
42. Some specific class meetings should not be scheduled in the same day.				✓						
43. The class meeting hours of any course should be scheduled in a consecutive format.	✓	✓			✓			✓		CPLEX(12)

Constraint	[1]	[2]	[4]	[5]	[6]	[7]	[10]	[11]	[12]	Our model
Class meeting										
44. The class meetings must be assigned to an available time slot.				✓						CPLEX(6)
45. The course should not be allocated to a time period inconvenient for heads of department because of other work.									✓	
46. The second time for each class meeting should not be in the same day.									✓	CPLEX(13)

Next, we will explain the parameters, decision variables, objective function, and constraints from research papers which are reasonable for our department.

2.2.1 Parameters

Let C be a set of class meetings,

R be a set of room types,

L be a set of lecturers,

S be a set of groups of students,

T be a set of time slots,

T_{slot} be a set of unavailable time slots,

D be a set of days,

D_d be a set of time slots in day d,

C_r be class meetings requiring a room type r, $\forall r \in R$,

C_l be class meetings that are taught by lecturer l, $\forall l \in L$,

C_s be class meetings of students in groups s, $\forall s \in S$,

T_l be a set of non-available times for each lecturer l, $\forall l \in L$,

T_r be a set of non-available times for each room r, $\forall r \in R$,

N_r be a number of available rooms of type r, $\forall r \in R$,

L_{max} be a maximum number of total class meetings a lecturer can teach in one day,

S_{max} be a maximum number of total class meetings a student is allowed to attend each day,

mc_l be a maximum consecutive classes a lecturer l is allowed to teach on the same day,

mc_s be a maximum consecutive classes a student group s is allowed to attend on the same day,

F be a set of class meetings in pairs (c_i, c_j) that can not be scheduled in the same day where $c_i, c_j \in C$,

K be a set of class meetings in pairs (c_i, c_j) that need to have a day off between two of the classes where $c_i, c_j \in C$,

H_c be a number of consecutive time slot per time for each class meeting c, $\forall c \in C$,

and $p_{c,t}$ be the preference of having class meeting c at time slot t.

2.2.2 Decision Variables

$$x_{c,t} = \begin{cases} 1 & \text{if a class of meeting } c \text{ is assigned in time slot } t, \\ 0 & \text{Otherwise.} \end{cases}$$

2.2.3 Objective Function

The objective function is to maximize class meetings that are assigned in the preference time slot.

$$\max \sum_{c} \sum_{t} p_{c,t} x_{c,t}.$$

2.2.4 Constraints

The constraints of research papers which are reasonable for our department.

1. All class meetings must be assigned to a time slot:

$$\textstyle\sum_{t \in T} x_{c,t} = 1, \quad \forall c \in C.$$

2. A lecturer must not teach more than one class meeting at a time:

$$\textstyle\sum_{c \in C_l} x_{c,t} \leq 1, \quad \forall l \in L, \forall t \in T.$$

3. The student scheduling conflicts:

$$\textstyle\sum_{c \in C_s} x_{c,t} \leq 1, \quad \forall s \in S, \forall t \in T.$$

4. Availability of lecturers:

$$\textstyle\sum_{t \in T_l} \sum_{c \in C_l} x_{c,t} = 0, \quad \forall l \in L.$$

5. The room limitation restrictions:

$$\textstyle\sum_{c \in C_r} x_{c,t} \leq N_r, \quad \forall r \in R, \forall t \in T.$$

6. Availability of time slots:

$$\sum_{t \in T_{slot}} x_{c,t} = 0, \quad \forall c \in C.$$

7. $x_{c,t}$ is binary decision variables:

$$x_{c,t} \in \{0, 1\} \quad \forall c \in C, \forall t \in T.$$

8. Maximum total class meetings that a lecturer can teach in a day:

$$\sum_{t \in D_d} \sum_{c \in C_l} x_{c,t} \leq L_{max}, \quad \forall d \in D, \forall l \in L.$$

9. Maximum total class meetings that a student can attend in a day:

$$\sum_{t \in D_d} \sum_{c \in C_s} x_{c,t} \leq S_{max}, \quad \forall d \in D, \forall s \in S.$$

10. Maximum consecutive class meetings for a lecturer:

$$\sum_{c \in C_l} (x_{c,t} + x_{c,t+1} + ... + x_{c,t+mc_l}) \leq mc_l, \quad \forall t \in \{t_1, t_2, ..., t_{td-mc_l}\}.$$

11. Maximum consecutive class meetings for a student:

$$\sum_{c \in C_s} (x_{c,t} + x_{c,t+1} + ... + x_{c,t+mc_s}) \leq mc_s, \quad \forall t \in \{t_1, t_2, ..., t_{td-mc_s}\}.$$

12. The class meeting hours of any course should be scheduled in a consecutive format:

$$x_{c,t} - x_{c,t+i} \leq 0, \quad \forall c \in C, \forall d \in D, \forall h \in H_c, \forall i \in \{1, 2, ..., h-1\}.$$

$$-x_{c,t} + x_{c,t+1} - x_{c,t+j} \leq 0, \quad \forall c \in C, \forall d \in D, \forall h \in H_c, \forall j \in \{2, 3, ..., h-1\}.$$

13. Some class meetings should not be scheduled in the same day:

$$\sum_{t \in D_d} x_{c_i,t} - x_{c_j,t} \leq 1, \quad \forall d \in D, \forall (c_i, c_j) \in F.$$

14. Having a day off between two classes of the same course is another requirement often found in universities. Below is the formulation:

$$\sum_{t \in D_d} x_{c_i,t} + \sum_{t \in D_d + D_{d+1}} x_{c_j,t} \leq 1, \quad \forall d \in D, \forall (c_i, c_j) \in K.$$

We need to adjust parameters and constraints in Subsections 2.2.1 and 2.2.2 to be able to fit in the scope of our department practice. The details are provided in Chapter 3.

Chapter 3
Classroom scheduling problem with integer linear program

3.1 Data adjustment

For this project, we use data of the first-semester of 2019 provided by the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University and the Office of the registrar. There are two sets of information given by the department which are teaching timetable and available time slots of each room. We adjust these data for our model.

3.1.1 Time slots

In this project, the time slots are slots from Monday to Friday from 8 a.m. to 5 p.m. We set 1 time slot to be 30 minutes. Therefore, we have 18 time slots per day or 90 time slots per week. Moreover, we set the department meetings on Wednesday from 1 - 4 p.m. as the unavailable time slots.

3.1.2 Teaching timetable

From the teaching timetable of the department, there are 109 courses and 187 class meetings (class meetings represent courses with course sections). This teaching timetable data consist of a list of course ID, course title, course section,

teach type, meeting day, meeting time, room, lecturer, and student group as we can see in Figure 3.1. Appendix A provides the full details of Figure 3.1.

	A	B	C	D	E	F	G	H	I	J
1	ตารางสอน ภาคต้น ปีการศึกษา 2562									
2	Term	Catalog	Course	Sec	Tea	Meeting Day	Meeting	Room	Instructor	Remarks
3	1621	2301101	CALCULUS I	1	LEC	MO TU WE TH	11:00-12:00	MAHIT-407	วาสนา	STATISTICS
4	1621	2301101	CALCULUS I	2	LEC	MO WE	08:00-09:00	TAB-220	พิมพ์เพ็ญ,ณัฐกาญจน์	EDUCATION
5	1621	2301101	CALCULUS I	2	LEC	FR	10:00-12:00	TAB-222	พิมพ์เพ็ญ,ณัฐกาญจน์	EDUCATION
6	1621	2301103	CALCULUS I	1	LEC	TU	10:00-12:00	TAB-221	รดินันท์,อธิปัตย์	PHARM SCI
7	1621	2301103	CALCULUS I	1	LEC	TH	11:00-12:00	TAB-221	รดินันท์,อธิปัตย์	PHARM SCI
8	1621	2301103	CALCULUS I	2	LEC	TU	10:00-12:00	TAB-222	เอื้อมพร	PHARM CARE
9	1621	2301103	CALCULUS I	2	LEC	TH	11:00-12:00	TAB-222	เอื้อมพร	PHARM CARE
10	1621	2301103	CALCULUS I	3	LEC	MO	09:00-10:00	TAB-222	ณัฏฐนาถ	OCARE
11	1621	2301103	CALCULUS I	3	LEC	WE	09:00-10:00	TAB-222	ณัฏฐนาถ	OCARE
12	1621	2301103	CALCULUS I	3	LEC	TH	08:00-10:00	TAB-220	ณัฏฐนาถ	OCARE
13	1621	2301107	CALCULUS I	1	LEC	WE	08:00-09:30	ENG1-113	วิมลรัตน์	ENGINEERING
14	1621	2301107	CALCULUS I	1	LEC	FR	08:00-09:30	ENG1-113	วิมลรัตน์	ENGINEERING
15	1621	2301107	CALCULUS I	2	LEC	WE	08:00-09:30	ENG3-315	สำรวม	ENGINEERING
16	1621	2301107	CALCULUS I	2	LEC	FR	08:00-09:30	ENG3-315	สำรวม	ENGINEERING
17	1621	2301107	CALCULUS I	3	LEC	MO	08:00-09:30	ENG1-302	ยศนันต์	ENGINEERING
18	1621	2301107	CALCULUS I	3	LEC	WE	08:00-09:30	ENG1-302	ยศนันต์	ENGINEERING
19	1621	2301107	CALCULUS I	4	LEC	MO	09:30-11:00	ENG1-302	กีรติ	ENGINEERING
20	1621	2301107	CALCULUS I	4	LEC	WE	09:30-11:00	ENG1-302	กีรติ	ENGINEERING
21	1621	2301107	CALCULUS I	5	LEC	MO	09:30-11:00	ENG3-113	ไพศาล	ENGINEERING
22	1621	2301107	CALCULUS I	5	LEC	FR	09:30-11:00	ENG3-113	ไพศาล	ENGINEERING
23	1621	2301107	CALCULUS I	6	LEC	MO	09:30-11:00	ENG3-409	บุญฤทธิ์,วุฒิชัย	ENGINEERING
24	1621	2301107	CALCULUS I	6	LEC	FR	09:30-11:00	ENG3-409	บุญฤทธิ์,วุฒิชัย	ENGINEERING
25	1621	2301107	CALCULUS I	7	LEC	MO WE FR	09:00-10:00	TAB-221	อนุสรณ์	CHEMTECH
26	1621	2301113	CALCULUS I	1	LEC	MO WE FR	08:00-09:00	MHMK-201	วาสนา,ศจี	BIO,ENVI
27	1621	2301113	CALCULUS I	1	LEC	TU	08:00-09:00	MHMK-M01	วาสนา,ศจี	BIO,ENVI
28	1621	2301113	CALCULUS I	2	LEC	MO WE FR	10:00-11:00	MHMK-M02	ธีรพงษ์	BIO,ENVI
29	1621	2301113	CALCULUS I	2	LEC	TU	08:00-09:00	TAB-220	ธีรพงษ์	BIO,ENVI
30	1621	2301115	CALCULUS	1	LEC	MO	10:00-12:00	MAHIT-401	พัฒนี	MANAGEMENT
31	1621	2301115	CALCULUS	1	LEC	WE	10:00-11:00	MAHIT-401	พัฒนี	MANAGEMENT
32	1621	2301115	CALCULUS	2	LEC	MO	10:00-12:00	MAHIT-402	พิเชฐ	MANAGEMENT

ภาคต้น 62

Figure 3.1: The teaching timetable created by of the department

These data are adjusted as follows:

Step 1: In this project, we focus on

(i) The courses assigned in a classroom in Maha Vajirunhis, Mahamakut, and Tab Nilaniti buildings.

(ii) The courses taught on Monday to Friday from 8 a.m. to 5 p.m.

(iii) The courses having lectures and laboratories.

So we ignore other courses that do not have the above qualifications. In the course "SEMINAR" section 2, we split into 3 class meetings: "2301490-21", "2301490-22", and "2301490-23" because we practically split into 3 classrooms. Therefore our project has 72 courses and 106 class meetings.

Step 2: From the previous step, the lecturers who teach the ignored courses will have the corresponding time slots of the ignored courses assigned as busy. Figure 3.2 shows that 1 means lecturer l is busy at time slot t and 0 means lecturer l is available at time slot t, for example teacher Amorn cannot teach at time slots 1 to 3 (8.00 - 9.30 a.m.) but he could be assigned to teach at time slot 4. Reader could see in Appendix C at Sheet "T_l" for more details.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
1	1=busy	Monday																		Tuesday																	
2	**Lecturer**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	Amorn	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Anusorn	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Arthorn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Athipat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Boonyarit	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	Chariya	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
9	Chatchawit	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Chitchuab	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	Chotiros	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Dittaya	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	Dumrong	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	Jaruloj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	Jiraphan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	Keng	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	Khamron	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	Kirati	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	Kitiporn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Kritsana	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	Krung	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Monchai	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	Monnat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	Nagul	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	Nataphan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	Nattakarn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	Nattanard	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	Ouamporn	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	Pakawan	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	Patanee	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	Pattarasinee	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	Peraphon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.2: Non-available times for each lecturer

Step 3: We add lists of students and the data of number of students for each class meeting from the Office of the registrar.

Step 4: From the previous step, we had the data of number of students for each class meeting, hence we can assign room type for each class meeting. The room types are divided into 4 types: Small (less than 40 seats), Medium (between 40 and 90 seats), Large (more than 90 seats), and Laboratory type.

Step 5: We assign C_{ij}, representing a set of the class meetings using j time slots per lecture and repeating i times per week. Finally, we can see the rearranged data for our CPLEX program in Figure 3.3. Appendix C at Sheet "data" provides the full version of Figure 3.3.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Course name	Course ID	ection	Class meeting	C_i	Lecturers		Students				#S	Room rec
2	CALCULUS I	2301101	2	2301101-2	2x	Pimpen	Nattakarr	EDU01				103	Large
3	CALCULUS I	2301103	1	2301103-1	2x	Ratinan	Athipat	PHARMSCI01				98	Large
4	CALCULUS I	2301103	2	2301103-2	2x	Ouamporn		PHARMCARE01				100	Large
5	CALCULUS I	2301103	3	2301103-3	3x	Nattanard		SAR01				61	Medium
6	CALCULUS I	2301107	7	2301107-7	32	Anusorn		CHEMTECH01				89	Medium
7	CALCULUS I	2301113	1	2301113-1	42	Vasana	Sajee	BIO01	ENVI01			138	Large
8	CALCULUS I	2301113	2	2301113-2	42	Teeraphong		BIO01	ENVI01			178	Large
9	CALCULUS I	2301117	1	2301117-1	42	Surachai		COMP01	PHOTO0	MATH01		184	Large
10	CALCULUS I	2301117	2	2301117-2	42	Monchai		PHYS01	MATSC01	GEO01		173	Large
11	CALCULUS I	2301117	3	2301117-3	42	Yuwaree		CHEM01				118	Large
12	CALCULUS I	2301117	4	2301117-4	42	Nattanard		FOOD01	MARINE01			123	Large
13	COMP PROG	2301170	1	2301170-1	2x	Dittaya		COMP01				81	Medium
14	COMP PROG	2301170	2	2301170-2	2x	Chotiros		GEO01	FOOD01	BIOCHEM	MICRO01	119	Large
15	COMP PROG	2301170	3	2301170-3	32	Sasipa		PHOTO0	MARINEC	MATSC01	PHYS01	157	Large
16	COMP PROG	2301170	4	2301170-4	32	Jaruloj		BIO01	BOT01	GENET01	MATH01	111	Large
17	COMP PROG	2301172	L001	2301172-L001	14	Jaruloj		MATH01				Lab	Lab
18	COMP PROG	2301172	L002	2301172-L002	14	Jaruloj		COMP01				Lab	Lab
19	COMP PROG	2301172	L003	2301172-L003	14	Jaruloj		MATSC01				Lab	Lab
20	COMP PROG	2301172	L004	2301172-L004	14	Jaruloj		MARINEC	BIOCHEM01			Lab	Lab
21	FIN MATH	2301181	1	2301181-1	2x	Kritsana	Tippawar	MATH01	EDU01			61	Medium
22	FIN MATH	2301181	2	2301181-2	2x	Kritsana	Tippawar	EDU01				44	Medium
23	CALCULUS III	2301203	2	2301203-2	32	Yuwaree		EDU02				52	Medium
24	CALCULUS III	2301217	1	2301217-1	2x	Amorn		COMP02				89	Medium
25	CALCULUS III	2301217	2	2301217-2	42	Surachai		MATH02				30	Small
26	PRIN MATH I	2301221	1	2301221-1	33	Chitchuab		EDU02				74	Medium
27	PRIN MATH	2301224	1	2301224-1	3x	Athipat		MATH02				30	Small
28	DISCRETE	2301232	1	2301232-1	24	Chariya	Teerapho	EDU02				33	Small
29	DISCRETE	2301233	1	2301233-1	24	Wutichai		SCIENCE				63	Medium
30	LINEAR	2301234	1	2301234-1	24	Yotsanan		MATH02				32	Small
31	GEOMETRY	2301242	1	2301242-1	2x	Nataphan						40	Small
32	PROB/STAT	2301247	1	2301247-1	2x	Tippawan		EDU02				37	Small

data | C | T_l | T_r | prefer | result

Figure 3.3: Rearranged data for CPLEX program

3.1.3 Room timetable

From the room timetable provided by faculty of Science (Figure 3.4 and full data in Appendix B), we adjust this timetable into our data. The time slots are changed from 1 hour into 30 minutes per slot. For each room, if it has an unrelated class meeting to our project, the associated time slots for this class meeting are busy. But if it has our department class meeting, the associated time slots are free. For example in Figure 3.4 at the cell D6, class meeting "2304103(1)" is unrelated to our project. Thus room "M01" in Mahamakut building at 8 - 9 a.m. (time slot 1 - 2) is busy because it is used for "2304103(1)". On the other hand, at the cell D13, class meeting "2301312(1)" is related to our project. It means that room "208" in Mahamakut building is free at 8 - 9 a.m. (time slot 1 - 2) because we will assign a new schedule to this time slot. We can see the modified room timetable in Figure 3.5 and can see in Appendix C at Sheet "T_r" for more details.

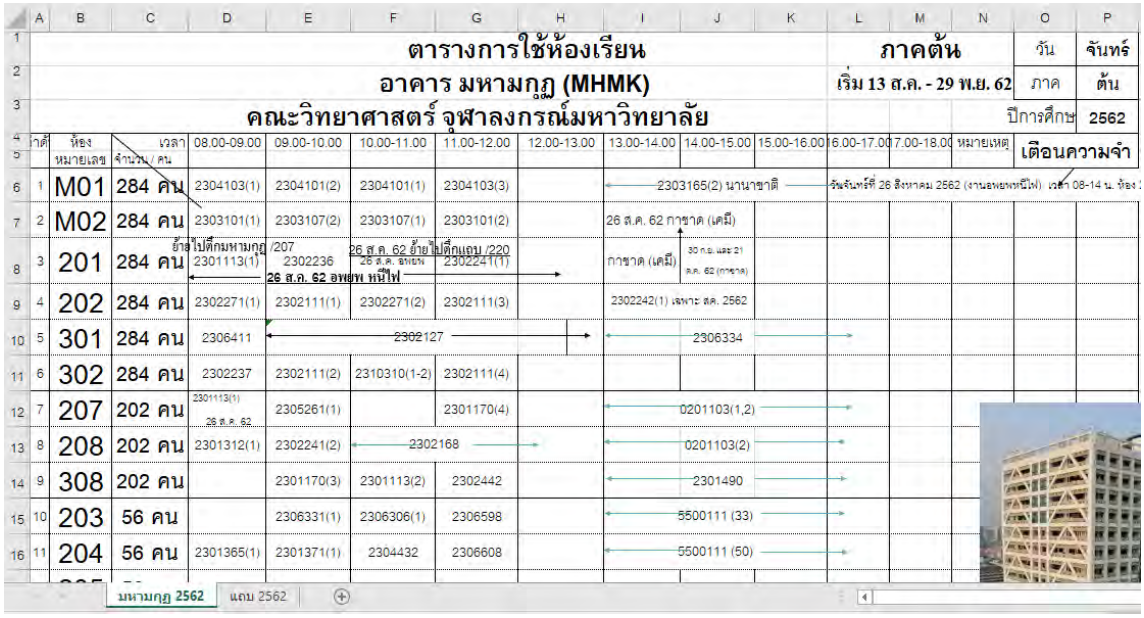
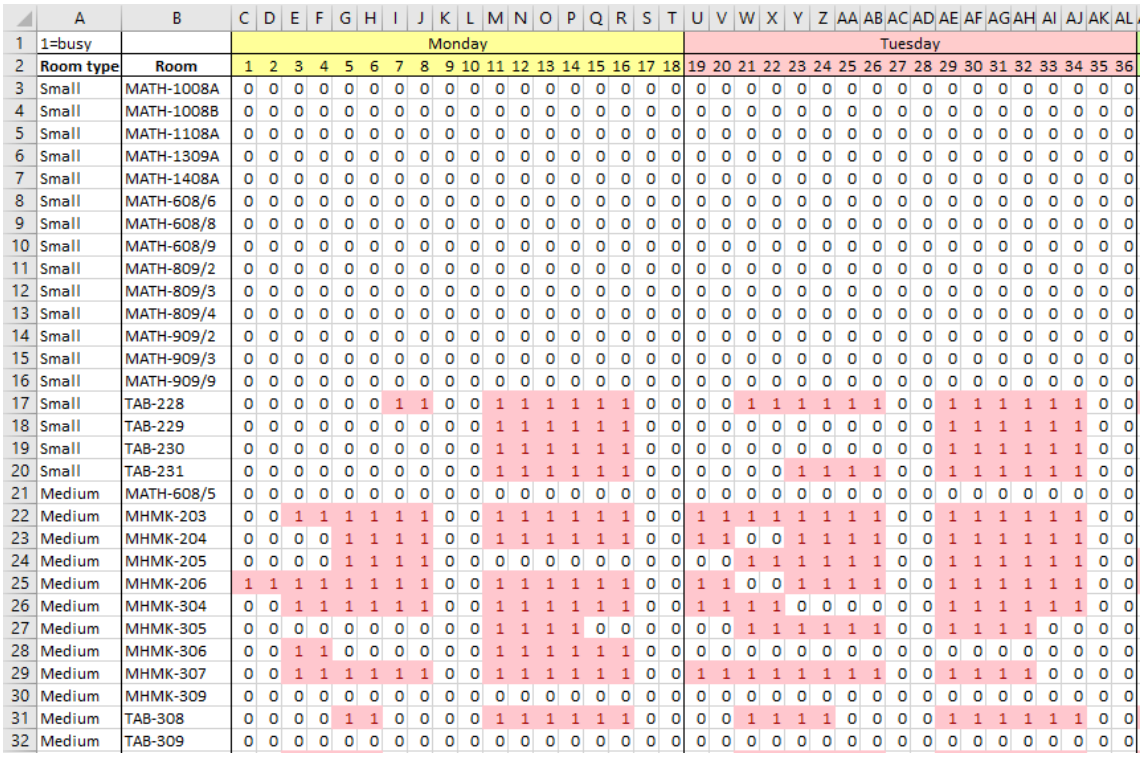


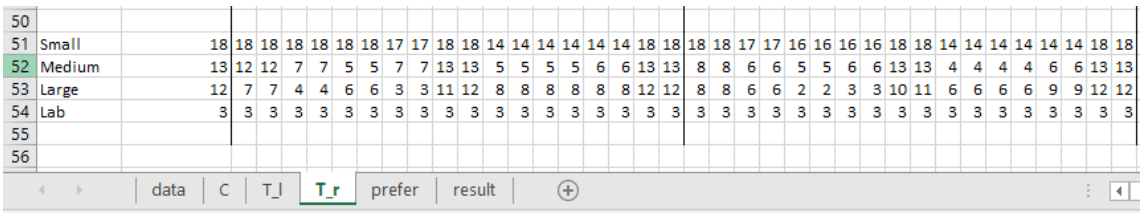
Figure 3.4: The faculty room timetable



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1	1=busy		Monday																		Tuesday																	
2	**Room type**	**Room**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	Small	MATH-1008A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Small	MATH-1008B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Small	MATH-1108A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Small	MATH-1309A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Small	MATH-1408A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	Small	MATH-608/6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	Small	MATH-608/8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Small	MATH-608/9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	Small	MATH-809/2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Small	MATH-809/3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	Small	MATH-809/4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	Small	MATH-909/2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	Small	MATH-909/3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	Small	MATH-909/9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	Small	TAB-228	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
18	Small	TAB-229	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
19	Small	TAB-230	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
20	Small	TAB-231	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
21	Medium	MATH-608/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Medium	MHMK-203	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
23	Medium	MHMK-204	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
24	Medium	MHMK-205	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
25	Medium	MHMK-206	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
26	Medium	MHMK-304	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
27	Medium	MHMK-305	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0
28	Medium	MHMK-306	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	Medium	MHMK-307	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0
30	Medium	MHMK-309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	Medium	TAB-308	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0
32	Medium	TAB-309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.5: Modified room timetable for CPLEX program

Next, we count the available room types r for each time slot t as Figure 3.6 and can see in Appendix C at Sheet "T_r" for more details.



50																																							
51	Small	18	18	18	18	18	18	18	17	17	18	18	14	14	14	14	14	14	18	18	18	18	17	17	16	16	16	16	18	18	14	14	14	14	14	14	18	18
52	Medium	13	12	12	7	7	5	5	7	7	13	13	5	5	5	5	6	6	13	13	8	8	6	6	5	5	6	6	13	13	4	4	4	4	6	6	13	13
53	Large	12	7	7	4	4	6	6	3	3	11	12	8	8	8	8	8	8	12	12	8	8	6	6	2	2	3	3	10	11	6	6	6	6	9	9	12	12
54	Lab	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
55																																							
56																																							

data | C | T_l | T_r | prefer | result

Figure 3.6: The available room types r for each time slot t

3.1.4 Preference

The objective function in this model is to maximize the total preference of assigning class meeting c at time slot t. We set the preference of having each class meeting between 9 - 12 a.m. and 1 - 4 p.m. every weekday as 1. It is written in Excel for the CPLEX program as Figure 3.7 where 1 in the figure means we want to assign

class meeting c at time slot t and 0 means we do not want to assign class meeting c at time slot t. Reader can see in Appendix C at Sheet “prefer” for more details.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
1	1=assigned	Monday																		Tuesday																	
2	**Class meeting**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	2301101-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
4	2301103-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
5	2301103-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
6	2301103-3	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
7	2301107-7	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
8	2301113-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
9	2301113-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
10	2301117-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
11	2301117-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
12	2301117-3	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
13	2301117-4	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
14	2301170-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
15	2301170-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
16	2301170-3	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
17	2301170-4	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
18	2301172-L001	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
19	2301172-L002	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
20	2301172-L003	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
21	2301172-L004	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
22	2301181-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
23	2301181-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
24	2301203-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
25	2301217-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
26	2301217-2	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
27	2301221-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
28	2301224-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
29	2301232-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
30	2301233-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
31	2301234-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
32	2301242-1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0

Figure 3.7: The preference of assigning class meeting c at time slot t

We now can apply our modify data to our model in a similar way as in Subsections 2.2.1 - 2.2.4, as follows.

3.2 Mathematical model

3.2.1 Parameters

Let C be a set of class meetings that contains course id “2301xxx” and course section “y” in the form of “2301xxx-y”, e.g. {2301101-2, 2301103-1, 2301103-2, ...},

R be a set of room types which is divided into 4 types: Small, Medium, Large, and Lab, i.e. {Small, Medium, Large, Lab},

L be a set of lecturers, e.g. {Amorn, Anusorn, Arthorn, ...},

S be a set of groups of students with the last two numbers represent first-year, second-year, third-year, fourth-year, respectively, e.g. {BIO01, BIOCHEM01, CHEMTECH02, ...},

D be a set of weekdays, i.e. {Mon, Tue, Wed, Thu, Fri},

T be a set of time slots from Monday to Friday 8 a.m. to 5 p.m., 1 time slot represents 30 minutes with the total of 90 time slots, i.e. {1, 2, 3, ..., 90},

T^b be a set of unavailable (busy) time slots, which is Wednesday 1 p.m. to 4 p.m., i.e. {47, 48, 49, 50, 51, 52},

T_d be a set of time slots for day d, e.g. T_{mon} = {1, 2, 3, ..., 18}, T_{tue} = {19, 20, 21, ..., 36},

C_{ij} be a set of class meetings using j time slots per lecture and repeating i times per week, e.g. C_{22} represents a set of class meetings using 2 time slots per lecture and repeating 2 times per week,

C_r be a set of class meetings requiring a room type r, $\forall r \in R$,

C_l be a set of class meetings taught by lecturer l, $\forall l \in L$,

C_s be a set of class meetings of students in groups s, $\forall s \in S$,

$L^b{}_{l,t}$ be a set of unavailable (busy) time slots of lecturer l, $\forall l \in L, \forall t \in T$,

$R^a{}_{r,t}$ be a set of the number of available room of types r at time slots t, $\forall r \in R, \forall t \in T$,

$p_{c,t}$ be the preference of having class meeting c at time slot t.

3.2.2 Decision Variables

Let $x_{c,t}$ be an integer number and

$$x_{c,t} = \begin{cases} 1, & \text{if a class of meeting } c \text{ is assigned in time slot } t, \\ 0, & \text{otherwise.} \end{cases}$$

3.2.3 Objective Function

The objective function is to maximize class meetings that are assigned in the preference time slot.

$$\max \sum_{c \in C} \sum_{t \in T} p_{c,t} x_{c,t}.$$

3.2.4 Constraints

The constraints can be written as mathematical relations and OPL language as follows:

1. **Constraint:** Each class meeting must be assigned to some time slots. They are divided into the class meetings study 2, 4, 6, 8, and 9 time slots per week.

   1.1 **Mathematical language:**

$$\sum_{t \in T} x_{c,t} = 2, \quad \forall c \in C_{12}$$

   OPL language: forall(c in C[“12”]: c in C) sum(t in T) x[c][t] == 2;

   1.2 **Mathematical language:**

$$\sum_{t \in T} x_{c,t} = 4, \quad \forall c \in C_{14}, C_{22}$$

   OPL language: forall(i in “14”, “22”, c in C[i]: c in C) sum(t in T) x[c][t] == 4;

   1.3 **Mathematical language:**

$$\sum_{t \in T} x_{c,t} = 6, \quad \forall c \in C_{16}, C_{23}, C_{2x}, C_{32}$$

OPL language: forall(i in “16”, “23”, “2x”, “32”, c in C[i]: c in C) sum(t in T) x[c][t] == 6;

Note that C_{2x} refers to class meetings that the department assigned to teach 2 times per week: 1 hour and 2 hours. We assign class meetings in C_{2x} as classes that teach 2 times per week with 1.5 hours each time instead, for convenience.

1.4 **Mathematical language:**

$$\sum_{t \in T} x_{c,t} = 8, \quad \forall c \in C_{18}, C_{24}, C_{3x}, C_{42}$$

OPL language: forall(i in “18”, “24”, “3x”, “42”, c in C[i]: c in C) sum(t in T) x[c][t] == 8;

Note that C_{3x} refers to class meetings that the department assigned to teach 3 times per week: 1 hour two times and 2 hours. We assign class meetings in C_{3x} as classes that teach 2 times per week with 2 hours each time instead, for convenience.

1.5 **Mathematical language:**

$$\sum_{t \in T} x_{c,t} = 9, \quad \forall c \in C_{33}$$

OPL language: forall(c in C[“33”]: c in C) sum(t in T) x[c][t] == 9;

2. **Constraint:** The lecturers must not teach more than one class meeting at a time. For each time slot, each lecturer has a choice between teach and free time.

   Mathematical language:

$$\sum_{c \in C_l} x_{c,t} \leq 1, \quad \forall l \in L, \forall t \in T$$

   OPL language: forall(l in L, t in T) sum(c in C[l]: c in C) x[c][t] <= 1;

3. **Constraint:** The students must not study more than one class meeting at a time. As same as the above constraint, each group student has a choice between study and free time for each time slot.

   Mathematical language:

$$\sum_{c \in C_s} x_{c,t} \leq 1, \quad \forall s \in S, \forall t \in T$$

OPL language: forall(s in S, t in T) sum(c in C[s]: c in C) x[c][t] <= 1;

4. **Constraint:** In our model, we have the class meetings that are required to teach by the lecturer l and a set of unavailable time slots of lecturer l. So if ${L^b}_{l,t}$ as 1 (lecturer l is busy at time slot t), $x_{c,t}$ should be 0 (class meeting c should be not assigned at time slot t).
   Mathematical language:

$$\sum_{t \in T} \sum_{c \in C_l} {L^b}_{l,t} * x_{c,t} = 0, \quad \forall l \in L$$

   OPL language: forall(l in L) sum(t in T, c in C[l]: c in C) L^b[l][t] * x[c][t] == 0;

5. **Constraint:** We have class meetings requiring rooms of type r and a set of the number of available rooms of type r at time slot t. So for each time slot, the number of class meetings requiring rooms of type r must be less than or equal to the number of available rooms of type r.
   Mathematical language:

$$\sum_{c \in C_r} x_{c,t} \leq {R^a}_{r,t}, \quad \forall r \in R, \forall t \in T$$

   OPL language: forall(r in R, t in T) sum(c in C[r]: c in C) x[c][t] <= R^a[r][t];

6. **Constraint:** All class meetings must be not assigned to unavailable time slots.
   Mathematical language:

$$\sum_{t \in T^b} x_{c,t} = 0, \quad \forall c \in C$$

   OPL language: forall(c in C) sum(t in T^b) x[c][t] == 0;

7. **Constraint:** $x_{c,t}$ is binary variable. Either a class meeting c is assigned in time slot t or a class meeting c is not assigned in time slot t.
   Mathematical language:

$$0 \leq x_{c,t} \leq 1, \quad \forall c \in C, \forall t \in T$$

OPL language: forall(c in C, t in T) 0 <= x[c][t] <= 1;

8. **Constraint:** Maximum total time slots that a lecturer can teach in a day is 12 time slots in a day. In our model, we found that each lecturer should not teach more than 12 time slots per day. Therefore we limit the maximum number of time slots for each lecturer.

   Mathematical language:

$$\sum_{t\in T_d}\sum_{c\in C_l} x_{c,t} \leq 12, \quad \forall l \in L, \forall d \in D$$

   OPL language: forall(l in L, d in D) sum(t in T[d]: t in T, c in C[l]: c in C) x[c][t] <= 12;

9. **Constraint:** Maximum total time slots that a student can attend in a day is 12 time slots in a day. As same as the above constraint, we found that each group student should not study more than 12 time slots per day.

   Mathematical language:

$$\sum_{t\in T_d}\sum_{c\in C_s} x_{c,t} \leq 12, \quad \forall s \in S, \forall d \in D$$

   OPL language: forall(s in S, d in D) sum(t in T[d]: t in T, c in C[s]: c in C) x[c][t] <= 12;

10. **Constraint:** We set the maximum consecutive time slots for a lecturer to be 8 time slots in a day, since the department assigns 8 time slots per lecture for class meeting “2301785-1”.

    Mathematical language:

$$\sum_{c\in C_l}\sum_{i=0}^{8} x_{c,t+i} \leq 8, \quad \forall l \in L, \forall t \in \{{T^1}_d, {T^2}_d, ..., {T^{10}}_d\}$$

   ${T^j}_d$ is j^{th} time slot in day d, $\forall d \in D$

   OPL language: forall(l in L, d in D, t in item(T[d], 0)..item(T[d], 9): t in T[d]) sum(c in C[l]: c in C) (x[c][t] + x[c][t+1] + ... + x[c][t+8]) <= 8;

11. **Constraint:** Maximum consecutive time slots for a student are 8 time slots in a day. The same as Constraint 10, groups of students of class meeting "2301785-1" have 8 consecutive time slots.

    Mathematical language:

    $$\sum_{c \in C_s} \sum_{i=0}^{8} x_{c,t+i} \leq 8, \quad \forall s \in S, \forall t \in \{T^1{}_d, T^2{}_d, ..., T^{10}{}_d\}$$

    $T^j{}_d$ is j^{th} time slot in day d, $\forall d \in D$

    OPL language: forall(s in S, d in D, t in item(T[d], 0)..item(T[d], 9): t in T[d]) sum(c in C[s] :c in C) (x[c][t] + x[c][t+1] + ... + x[c][t+8]) <= 8;

12. **Constraint:** Each lecture of a class meeting may have consecutive time slots. We have a set of the class meetings using j time slots per lecture and repeating i times per week, $i \in 1, 2, 3, 4$ and $j \in 2, 3, 4, 6, 8$. So the class meetings have repeating 1, 2, 3, 4 times per week. Case I: 1 time per week, these class meetings can be assigned to any day in the week. Case II: 2 times per week, these class meetings can choose to be assigned on (Monday and Wednesday) or (Tuesday and Thursday) or (Wednesday and Friday). Case III: 3 times per week, these class meetings should be assigned to Monday, Wednesday, and Friday. Case IV: 4 times per week, these class meetings should be assigned to Monday, Tuesday, Thursday, and Friday. Here, we illustrate only the case of C_{12}. Other cases can be seen at line 239 - 977 in Appendix D .

    Mathematical language:

    $(x_{c,t^1{}_{mon}} + x_{c,t^2{}_{mon}} = 2||x_{c,t^3{}_{mon}} + x_{c,t^4{}_{mon}} = 2||...||x_{c,t^{17}{}_{mon}} + x_{c,t^{18}{}_{mon}} = 2)||$

    $(x_{c,t^1{}_{tue}} + x_{c,t^2{}_{tue}} = 2||x_{c,t^3{}_{tue}} + x_{c,t^4{}_{tue}} = 2||...||x_{c,t^{17}{}_{tue}} + x_{c,t^{18}{}_{tue}} = 2)||$

    $(x_{c,t^1{}_{wed}} + x_{c,t^2{}_{wed}} = 2||x_{c,t^3{}_{wed}} + x_{c,t^4{}_{wed}} = 2||...||x_{c,t^{17}{}_{wed}} + x_{c,t^{18}{}_{wed}} = 2)||$

    $(x_{c,t^1{}_{thu}} + x_{c,t^2{}_{thu}} = 2||x_{c,t^3{}_{thu}} + x_{c,t^4{}_{thu}} = 2||...||x_{c,t^{17}{}_{thu}} + x_{c,t^{18}{}_{thu}} = 2)||$

    $(x_{c,t^1{}_{fri}} + x_{c,t^2{}_{fri}} = 2||x_{c,t^3{}_{fri}} + x_{c,t^4{}_{fri}} = 2||...||x_{c,t^{17}{}_{fri}} + x_{c,t^{18}{}_{fri}} = 2)||, \quad \forall c \in C_{12}$

    $T^j{}_d$ is j^{th} time slot in day d, $\forall d \in D$

    OPL language: forall(c in C["12"] :c in C)

    (x[c][item(T["mon"], 0)] + x[c][item(T["mon"], 1)] == 2 ||

    x[c][item(T["mon"], 2)] + x[c][item(T["mon"], 3)] == 2 ||

    ...|| x[c][item(T["mon"], 16)]+x[c][item(T["mon"], 17)] == 2)

||
(x[c][item(T[“tue”], 0)] + x[c][item(T[“tue”], 1)] == 2 ||
x[c][item(T[“tue”], 2)] + x[c][item(T[“tue”], 3)] == 2 ||
...|| x[c][item(T[“tue”], 16)]+x[c][item(T[“tue”], 17)] == 2)
||
(x[c][item(T[“wed”], 0)] + x[c][item(T[“wed”], 1)] == 2 ||
x[c][item(T[“wed”], 2)] + x[c][item(T[“wed”], 3)] == 2 ||
...|| x[c][item(T[“wed”], 16)]+x[c][item(T[“wed”], 17)] == 2)
||
(x[c][item(T[“thu”], 0)] + x[c][item(T[“thu”], 1)] == 2 ||
x[c][item(T[“thu”], 2)] + x[c][item(T[“thu”], 3)] == 2 ||
...|| x[c][item(T[“thu”], 16)]+x[c][item(T[“thu”], 17)] == 2)
||
(x[c][item(T[“fri”], 0)] + x[c][item(T[“fri”], 1)] == 2 ||
x[c][item(T[“fri”], 2)] + x[c][item(T[“fri”], 3)] == 2 ||
...|| x[c][item(T[“fri”], 16)]+x[c][item(T[“fri”], 17)] == 2);

13. **Constraint:** Class meetings should not have more than one lecture on the same day. By the above constraint, class meetings may repeat the lectures 1, 2, 3, and 4 times per week. So the next lecture for each class meeting should not be assigned on the same day.

13.1 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} = 0 || \sum_{t\in T_d} x_{c,t} = 2, \quad \forall c \in C_{12}, C_{22}, C_{32}, C_{42}, \forall d \in D$$

OPL language: forall(i in “12”,“22”,“32”,“42”, c in C[i]: c in C, d in D) sum(t in T[d]) x[c][t] == 0 || sum(t in T[d]) x[c][t] == 2;

13.2 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} = 0 || \sum_{t\in T_d} x_{c,t} = 4, \quad \forall c \in C_{14}, C_{24}, C_{3x}, \forall d \in D$$

OPL language:forall(i in “14”, “24”, “3x”, c in T[i]: c in C, d in D) sum(t in T[d]) x[c][t] == 0 || sum(t in T[d]) x[c][t] == 4;

13.3 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} = 0 || \sum_{t\in T_d} x_{c,t} = 3, \quad \forall c \in C_{23}, C_{2x}, C_{33}, \forall d \in D$$

OPL language:forall(i in “23”, “2x”, “33”, c in T[i]: c in C, d in D) sum(t in T[d]) x[c][t] == 0 || sum(t in T[d]) x[c][t] == 3;

13.4 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} = 0 || \sum_{t\in T_d} x_{c,t} = 6, \quad \forall c \in C_{16}, \forall d \in D$$

OPL language:forall(c in C[“16”] :c in C, d in D) sum(t in T[d]) x[c][t] == 0 || sum(t in T[d]) x[c][t] == 6;

13.5 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} = 0 || \sum_{t\in T_d} x_{c,t} = 8, \quad \forall c \in C_{18}, \forall d \in D$$

OPL language:forall(c in T[“18”] :c in C, d in D) sum(t in T[d]) x[c][t] == 0 || sum(t in T[d]) x[c][t] == 8;

14. **Constraint:** Class meetings should have a day break between 2 consecutive lectures. By the Constraint 12, we have been assigned about class meetings have repeating 2 and 3 times per week. This constraint ensures these class meetings were not assigned on the consecutive day.

14.1 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} + \sum_{t\in T_{d+1}} x_{c,t} <= 2, \quad \forall c \in C_{22}, C_{32}, \forall d \in \{mon, tue, wed, thu\}$$

OPL language: forall(i in “22”, “32”, c in C[i]: c in C, d in “mon”, “tue”, “wed”, “thu”) sum(t in T[d]) x[c][t] + sum(t in T[next(day, d)]) x[c][t] <= 2;

14.2 **Mathematical language:**

$$\sum_{t\in T_d} x_{c,t} + \sum_{t\in T_{d+1}} x_{c,t} <= 3, \quad \forall c \in C_{23}, C_{2x}, C_{33}, \forall d \in \{mon, tue, wed, thu\}$$

OPL language:forall(i in “23”, “2x”, “33”, c in C[i]: c in C, d in “mon”, “tue”, “wed”, “thu”) sum(t in T[d]) x[c][t] + sum(t in T[next(day, d)]) x[c][t] <= 3;

14.3 **Mathematical language:**

$$\sum_{t \in T_d} x_{c,t} + \sum_{t \in T_{d+1}} x_{c,t} <= 4, \quad \forall c \in C_{24}, C_{3x}, \forall d \in \{mon, tue, wed, thu\}$$

OPL language:forall(i in “24”, “3x”, c in C[i]: c in C, d in “mon”, “tue”, “wed”, “thu”) sum(t in T[d]) x[c][t] + sum(t in T[next(day, d)]) x[c][t] <= 4;

We can see that the number of time slots used in constraint 8 allows lecturer to teach 6 hours per day. Similarly, students may attend in classroom up to 6 hours per day. In the next chapter, we will discuss the possibility of reducing these hours and explain the limitation of our program.

Chapter 4
Programs and Results

In the previous chapter, we show our integer linear programming model. Next, we will show how to using CPLEX Studio IDE software version 12.6.3 and Python 3.7 to solve a problem, how to run the programs, and result from these programs.

4.1 CPLEX Studio IDE software version 12.6.3

4.1.1 Creating an OPL project

When you open CPLEX, the main window will appear as we can see in Figure 4.1. Then, we will create our OPL project by click “New” and “OPL project” buttons as shown in Figure 4.2. The program will show the “New project” window. In this window, we will create a project name, select project location, click create model and create data and click the “Finish” button as presented in Figure 4.3. Then, the OPL project will appear and we can write the OPL code as presented in the previous chapter as in Figure 4.4. We write our code in 2 files: Classroom Scheduling.mod (write OPL code in Chapter 3) and Classroom Scheduling.dat (write OPL code to call data from Excel for Classroom Scheduling.mod).

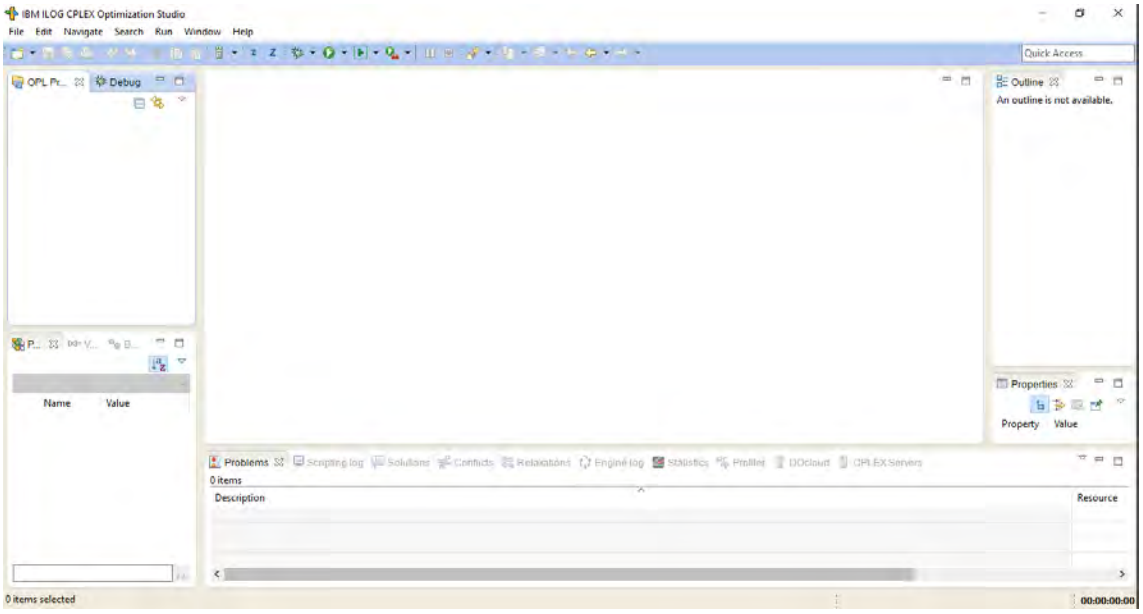


Figure 4.1: The main window of CPLEX Studio IDE software version 12.6.3

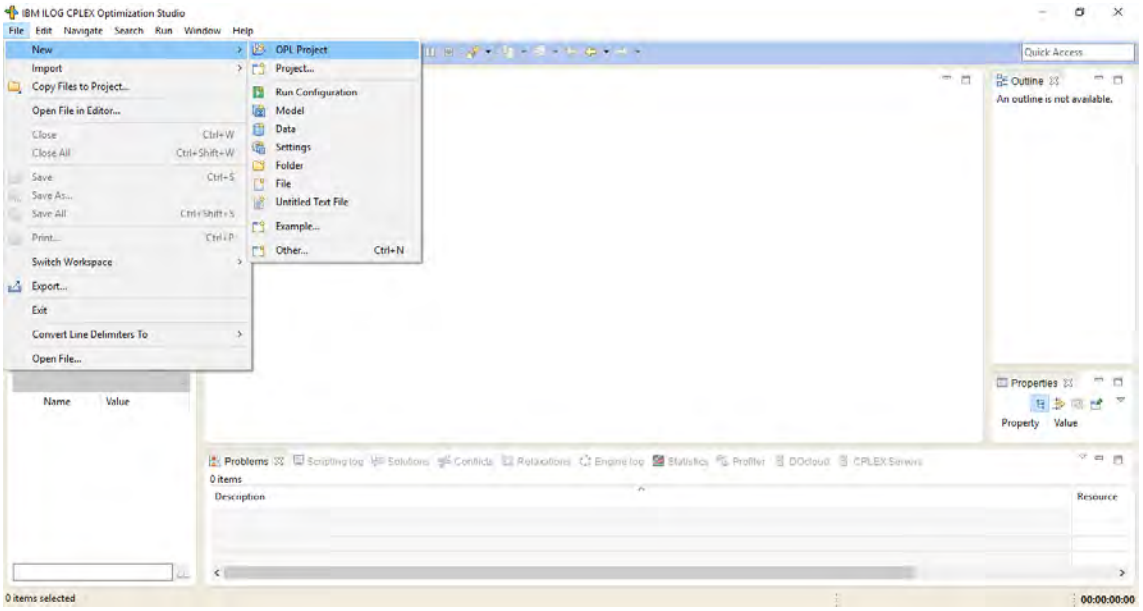


Figure 4.2: Creating a new project

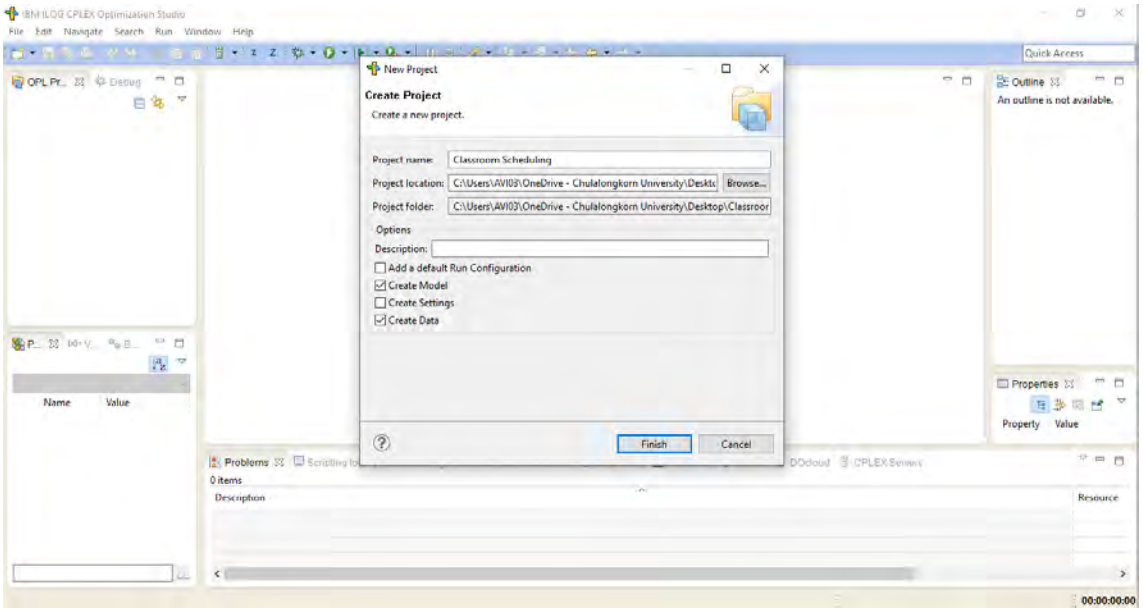


Figure 4.3: Creating a new project (Cont.)

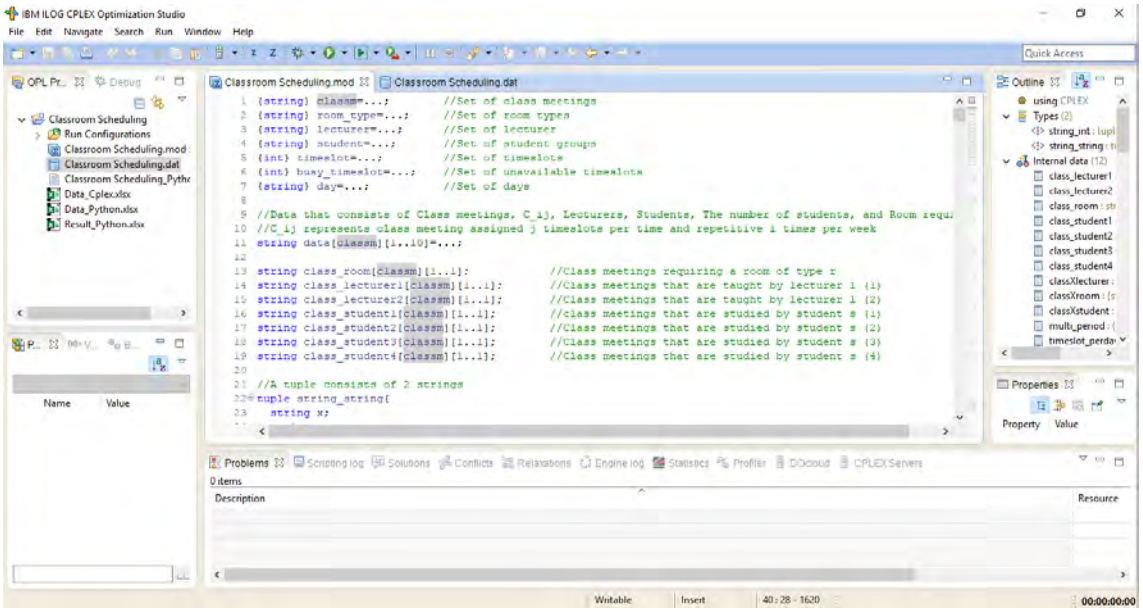


Figure 4.4: Classroom Scheduling.mod

4.1.2 Running our OPL project and result

We first write OPL project until line 104 in Classroom Scheduling.mod and line 21 in Classroom Scheduling.dat (see Appendix D), then run the CPLEX program which automatically create a set of class meetings requiring a room type r, a set of class meetings taught by lecturer l, and a set of class meetings having the same group student s in sheet "C" in Data_Cplex.xlsx.

Now we have full data in Excel. Next, we write and run complete code. Finally, the objective value (the total result of the objective function) is 643 as in Figure 4.5 and the result is appear in sheet "result" in Data_Cplex.xlsx as Figure 4.6 where 1 in the figure means class meeting c is assigned at time slot t and 0 means class meeting c is not assigned at time slot t. Reader could see full data at Sheet "result" in Appendix C.

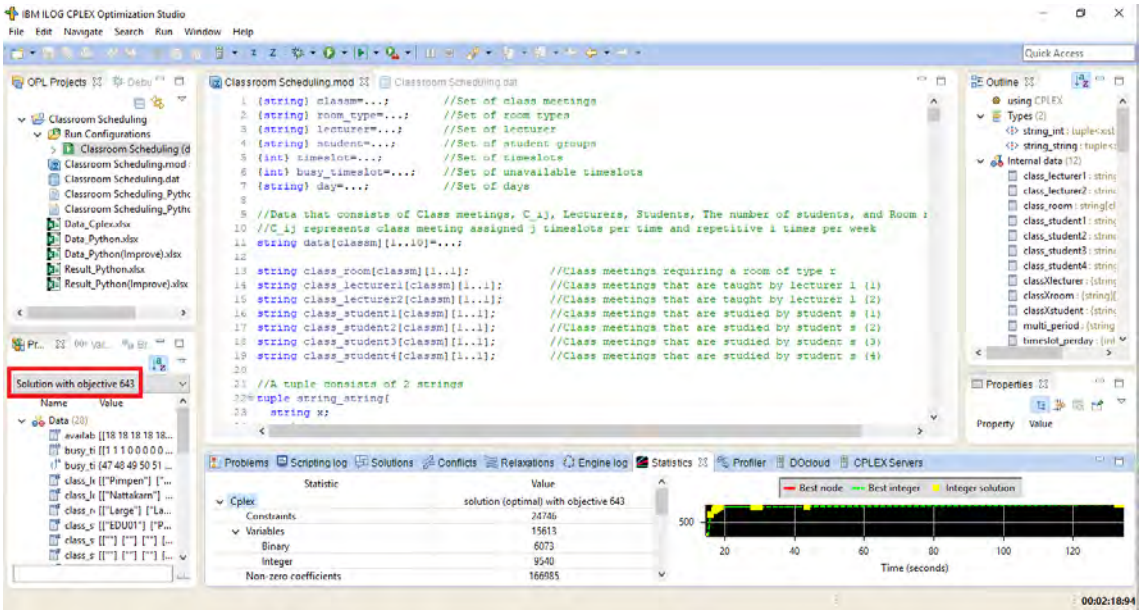


Figure 4.5: The result from CPLEX

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL
1		1=assigned	Monday																		Tuesday																	
2	C_ij	Class meeting	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	2x	2301101-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
4	2x	2301103-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
5	2x	2301103-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
6	3x	2301103-3	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	32	2301107-7	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	42	2301113-1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
9	42	2301113-2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	42	2301117-1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
11	42	2301117-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	42	2301117-3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	42	2301117-4	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
14	2x	2301170-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
15	2x	2301170-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	32	2301170-3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	32	2301170-4	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	14	2301172-L001	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	14	2301172-L002	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	14	2301172-L003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	14	2301172-L004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	2x	2301181-1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	2x	2301181-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
24	32	2301203-2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	2x	2301217-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
26	42	2301217-2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	33	2301221-1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	3x	2301224-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
29	24	2301232-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
30	24	2301233-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
31	24	2301234-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
32	2x	2301242-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
33	2x	2301247-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
34	2x	2301251-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
35	2x	2301261-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
36	2x	2301263-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.6: The result from CPLEX in Excel

Next, we use available rooms at time slots t and the result from Data_Cplex.xlsx as data in our Python to assign the room for each class meeting.

4.2 Python 3.7 and result

We get the class meeting c assigned at time slot t from CPLEX program. In the Python program, we will assign room for each class meeting. We present algorithm as a flowchart of our Python program in Figure 4.7 and the code is shown in Appendix F.

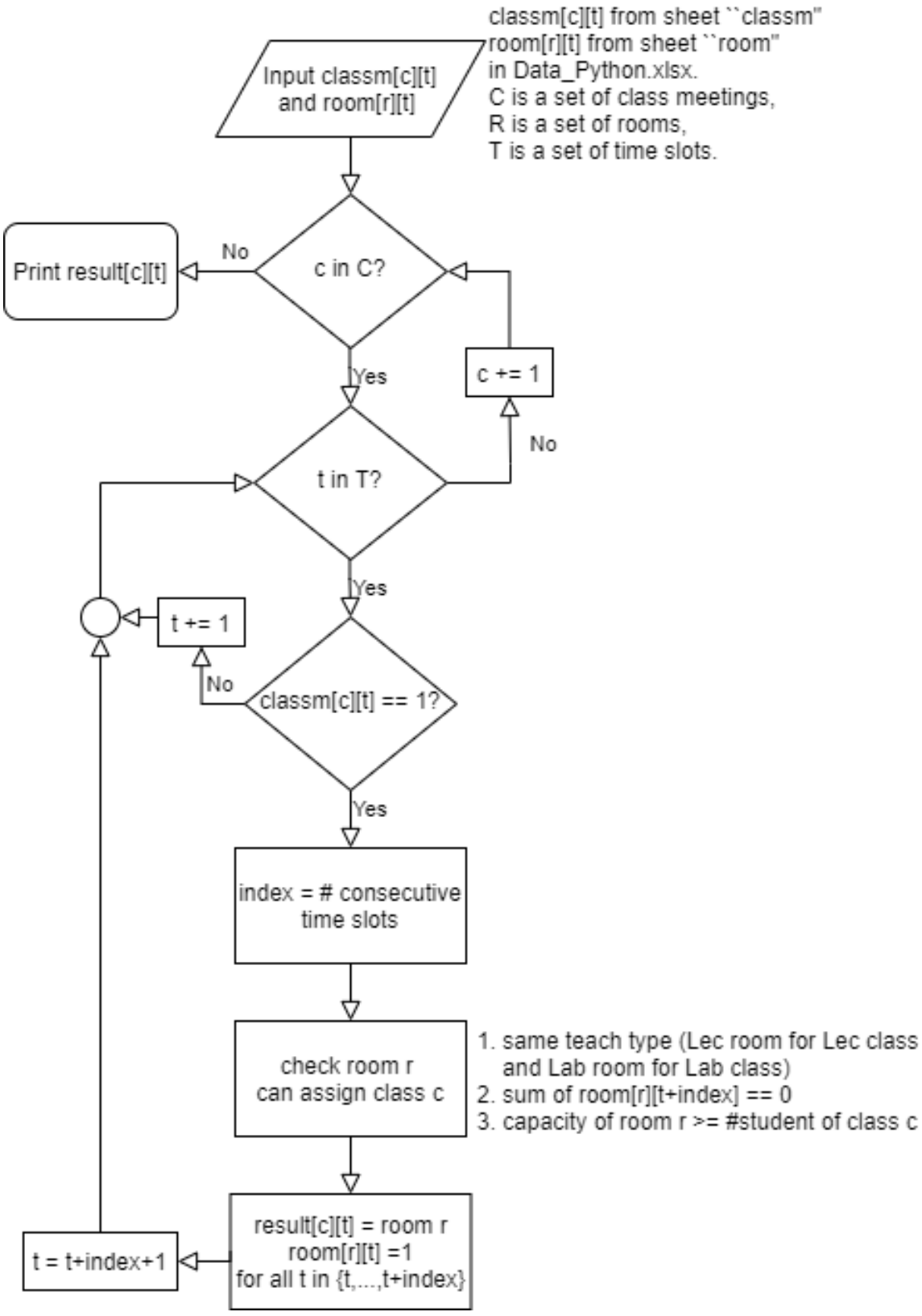


Figure 4.7: Flowchart of Python algorithm

In Python part, we create data for Python program in Data_Python.xlsx. This Excel consists of 2 sheets: 1.) “room” presents capacity, teaching type, and available time slot for each room (Figure 4.8) and 2.) “classm” presents lecturer, number of students, teaching type, and teaching time slots for each class meeting (Figure 4.9). Reader can see the full data in Appendix E.

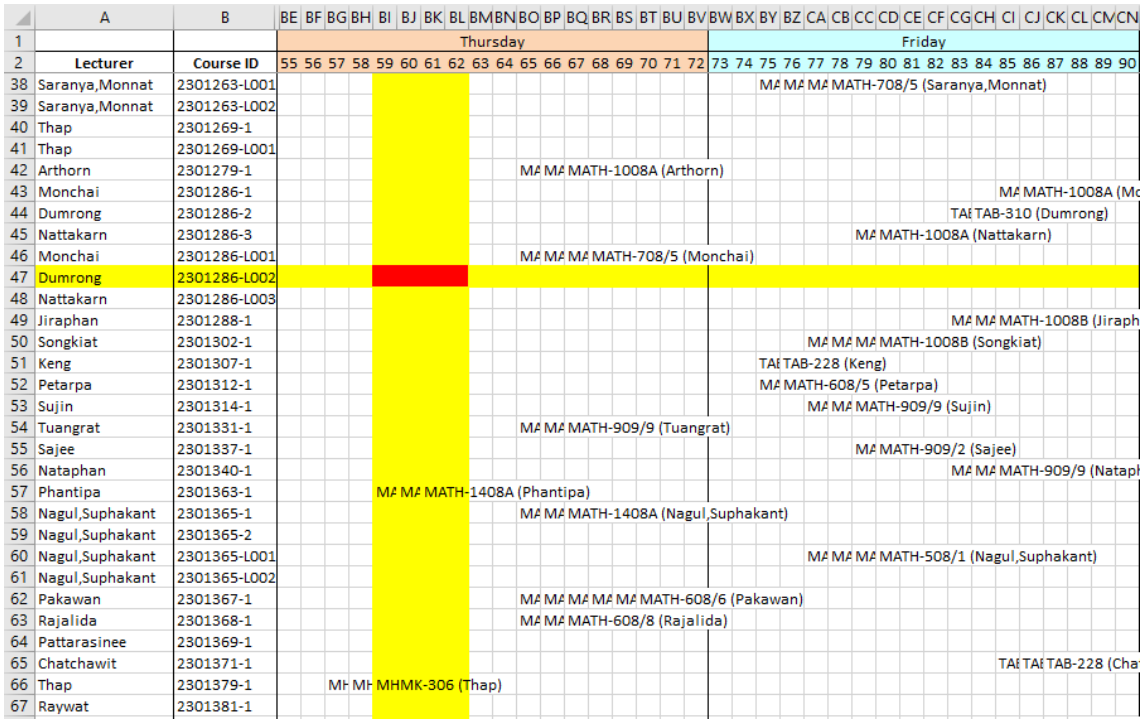
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM
1				Monday																		Tuesday																	
2	Capac	Teach	Room	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	30	Lec	MATH-1008A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	30	Lec	MATH-1008B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	30	Lec	MATH-1108A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	30	Lec	MATH-1309A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	20	Lec	MATH-1408A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	20	Lec	MATH-608/6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	20	Lec	MATH-608/8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	20	Lec	MATH-608/9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	30	Lec	MATH-809/2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	30	Lec	MATH-809/3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	30	Lec	MATH-809/4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	35	Lec	MATH-909/2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	30	Lec	MATH-909/3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	15	Lec	MATH-909/9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	40	Lec	TAB-228	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
18	40	Lec	TAB-229	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
19	40	Lec	TAB-230	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0
20	40	Lec	TAB-231	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
21	60	Lec	MATH-608/5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	56	Lec	MHMK-203	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
23	56	Lec	MHMK-204	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
24	56	Lec	MHMK-205	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
25	56	Lec	MHMK-206	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
26	56	Lec	MHMK-304	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
27	56	Lec	MHMK-305	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0
28	56	Lec	MHMK-306	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	56	Lec	MHMK-307	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0
30	56	Lec	MHMK-309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	90	Lec	TAB-308	0	0	0	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0
32	64	Lec	TAB-309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4.8: Sheet “room” in Data_Python.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN
1					Monday																		Tuesday																	
2	Lecturer	#S	Teac	Class meeting	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
3	Pimpen,Nattakarn	103	Lec	2301101-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
4	Ratinan,Athipat	98	Lec	2301103-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
5	Ouamporn	100	Lec	2301103-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
6	Nattanard	61	Lec	2301103-3	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Anusorn	89	Lec	2301107-7	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	Vasana,Sajee	138	Lec	2301113-1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
9	Teeraphong	178	Lec	2301113-2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	Surachai	184	Lec	2301117-1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
11	Monchai	173	Lec	2301117-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Yuwaree	118	Lec	2301117-3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	Nattanard	123	Lec	2301117-4	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
14	Dittaya	81	Lec	2301170-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
15	Chotiros	119	Lec	2301170-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	Sasipa	157	Lec	2301170-3	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	Jaruloj	111	Lec	2301170-4	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	Jaruloj	33	Lab	2301172-L001	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	Jaruloj	74	Lab	2301172-L002	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	Jaruloj	55	Lab	2301172-L003	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	Jaruloj	60	Lab	2301172-L004	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	Kritsana,Tippawan	61	Lec	2301181-1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	Kritsana,Tippawan	44	Lec	2301181-2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
24	Yuwaree	52	Lec	2301203-2	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	Amorn	89	Lec	2301217-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
26	Surachai	30	Lec	2301217-2	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	Chitchuab	74	Lec	2301221-1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	Athipat	30	Lec	2301224-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
29	Chariya,Teeraphong	33	Lec	2301232-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
30	Wutichai	63	Lec	2301233-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
31	Yotsanan	32	Lec	2301234-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
32	Nataphan	40	Lec	2301242-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0

Figure 4.9: Sheet “classm” in Data_Python.xlsx

After run Python, the result is shown in Result_Python.xlsx as in Figure 4.10 (Full data are shown Appendix H). The result has a problem at course ID “2301286-L002”, which has 55 students in the course but there is no room with the seats more than number of students in the course at time slots 59-62 (this course should be assigned). Figure 4.11 shows Lab type rooms: “MATH-508/1” is free, “MATH-509/2” is assigned “2301172-L003” (number of students is 55), and “MATH-708/5” is assigned “2301679-L001” (number of students is 12). We adjust the number of students from 55 to 45. Now the room for “2301286-L002” is assigned as in Figure 4.12. This situation happens because we did not category “Lab” room type as small lab, medium lab, and large lab. We will leave this adjustment as our further study. An improved code for Python program is shown in Appendix G and the improved result from Python program is shown in Appendix I.



	A	B	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN
1			Thursday																		Friday																		
2	Lecturer	Course ID	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
38	Saranya,Monnat	2301263-L001																					MA	MA	MA	MATH-708/5 (Saranya,Monnat)												
39	Saranya,Monnat	2301263-L002																																				
40	Thap	2301269-1																																				
41	Thap	2301269-L001																																				
42	Arthorn	2301279-1											MA	MA	MATH-1008A (Arthorn)																								
43	Monchai	2301286-1																															MA	MATH-1008A (Mc					
44	Dumrong	2301286-2																													TA	TAB-310 (Dumrong)						
45	Nattakarn	2301286-3																										MA	MATH-1008A (Nattakarn)									
46	Monchai	2301286-L001											MA	MA	MA	MATH-708/5 (Monchai)																							
47	Dumrong	2301286-L002																																				
48	Nattakarn	2301286-L003																																				
49	Jiraphan	2301288-1																													MA	MA	MATH-1008B (Jiraph					
50	Songkiat	2301302-1																							MA	MA	MA	MATH-1008B (Songkiat)										
51	Keng	2301307-1																					TA	TAB-228 (Keng)														
52	Petarpa	2301312-1																					MA	MATH-608/5 (Petarpa)														
53	Sujin	2301314-1																							MA	MA	MATH-909/9 (Sujin)											
54	Tuangrat	2301331-1											MA	MA	MATH-909/9 (Tuangrat)																								
55	Sajee	2301337-1																										MA	MATH-909/2 (Sajee)									
56	Nataphan	2301340-1																													MA	MA	MATH-909/9 (Nataph					
57	Phantipa	2301363-1					MA	MA	MATH-1408A (Phantipa)																															
58	Nagul,Suphakant	2301365-1											MA	MA	MATH-1408A (Nagul,Suphakant)																								
59	Nagul,Suphakant	2301365-2																																				
60	Nagul,Suphakant	2301365-L001																							MA	MA	MA	MATH-508/1 (Nagul,Suphakant)										
61	Nagul,Suphakant	2301365-L002																																				
62	Pakawan	2301367-1											MA	MA	MA	MA	MA	MATH-608/6 (Pakawan)																					
63	Rajalida	2301368-1											MA	MA	MATH-608/8 (Rajalida)																								
64	Pattarasinee	2301369-1																																				
65	Chatchawit	2301371-1																															TA	TA	TAB-228 (Cha			
66	Thap	2301379-1			MH	MH	MHMK-306 (Thap)																															
67	Raywat	2301381-1																																				

Figure 4.10: The result from Python when “2301286-L002” has 55 students

	A	B	C	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO
1				Thursday																		Friday																		
2	Capa	Teac	Room	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
21	60	Lec	MATH-608/5	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	56	Lec	MHMK-203	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
23	56	Lec	MHMK-204	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0
24	56	Lec	MHMK-205	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
25	56	Lec	MHMK-206	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0
26	56	Lec	MHMK-304	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
27	56	Lec	MHMK-305	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
28	56	Lec	MHMK-306	0	0	1	1	1	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	56	Lec	MHMK-307	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0
30	56	Lec	MHMK-309	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	90	Lec	TAB-308	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1	0	0
32	64	Lec	TAB-309	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	80	Lec	TAB-310	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0
34	284	Lec	MHMK-201	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
35	284	Lec	MHMK-202	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
36	202	Lec	MHMK-207	1	1	1	1	1	1	1	1	0	0	1	1	1	0	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
37	202	Lec	MHMK-208	1	1	1	1	1	1	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
38	284	Lec	MHMK-301	1	1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0
39	284	Lec	MHMK-302	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
40	202	Lec	MHMK-308	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
41	284	Lec	MHMK-M01	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
42	284	Lec	MHMK-M02	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
43	250	Lec	TAB-220	0	0	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
44	250	Lec	TAB-221	0	0	1	1	1	1	1	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
45	250	Lec	TAB-222	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	45	Lab	MATH-508/1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0
47	75	Lab	MATH-509/2	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
48	40	Lab	MATH-708/5	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
49																																							

Figure 4.11: Available room after run Python

	A	B	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN
1			Thursday																		Friday																		
2	Lecturer	Course ID	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
38	Saranya,Monnat	2301263-L001																					MA	MA	MA	MATH-708/5 (Saranya,Monnat)												
39	Saranya,Monnat	2301263-L002																																				
40	Thap	2301269-1																																				
41	Thap	2301269-L001																																				
42	Arthorn	2301279-1											MA	MA	MATH-1008A (Arthorn)																								
43	Monchai	2301286-1																															MA	MATH-1008A (Mc					
44	Dumrong	2301286-2																													TA	TAB-310 (Dumrong)						
45	Nattakarn	2301286-3																									MA	MATH-1008A (Nattakarn)										
46	Monchai	2301286-L001											MA	MA	MA	MATH-708/5 (Monchai)																							
47	Dumrong	2301286-L002					MA	MA	MA	MATH-509/2 (Dumrong)																														
48	Nattakarn	2301286-L003																																				
49	Jiraphan	2301288-1																													MA	MA	MATH-1008B (Jiraph					
50	Songkiat	2301302-1																							MA	MA	MA	MATH-1008B (Songkiat)										
51	Keng	2301307-1																					TA	TAB-228 (Keng)														
52	Petarpa	2301312-1																					MA	MATH-608/5 (Petarpa)														
53	Sujin	2301314-1																							MA	MA	MATH-909/9 (Sujin)											
54	Tuangrat	2301331-1											MA	MA	MATH-909/9 (Tuangrat)																								
55	Sajee	2301337-1																									MA	MATH-909/2 (Sajee)										
56	Nataphan	2301340-1																													MA	MA	MATH-909/9 (Nataph					
57	Phantipa	2301363-1					MA	MA	MATH-1408A (Phantipa)																														
58	Nagul,Suphakant	2301365-1											MA	MA	MATH-1408A (Nagul,Suphakant)																								
59	Nagul,Suphakant	2301365-2																																				
60	Nagul,Suphakant	2301365-L001																							MA	MA	MA	MATH-508/1 (Nagul,Suphakant)										
61	Nagul,Suphakant	2301365-L002																																				
62	Pakawan	2301367-1											MA	MA	MA	MA	MA	MATH-608/6 (Pakawan)																				
63	Rajalida	2301368-1											MA	MA	MATH-608/8 (Rajalida)																								
64	Pattarasinee	2301369-1																																				
65	Chatchawit	2301371-1																															TA	TA	TAB-228 (Cha			
66	Thap	2301379-1			MH	MH	MHMK-306 (Thap)																														
67	Raywat	2301381-1																																				

Figure 4.12: The improved result from Python when “2301286-L002” has 45 students

The final result can be seen in as Figure 4.13. We separate the result into daily results as in Figure 4.14.

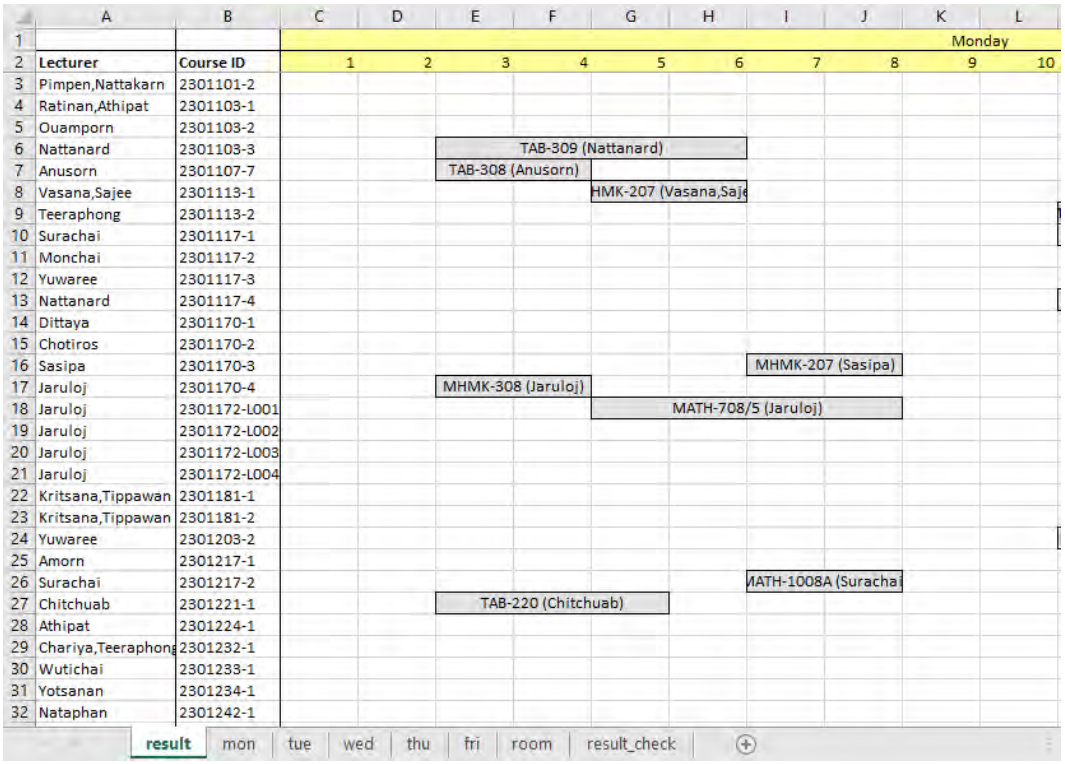


Figure 4.13: The final result

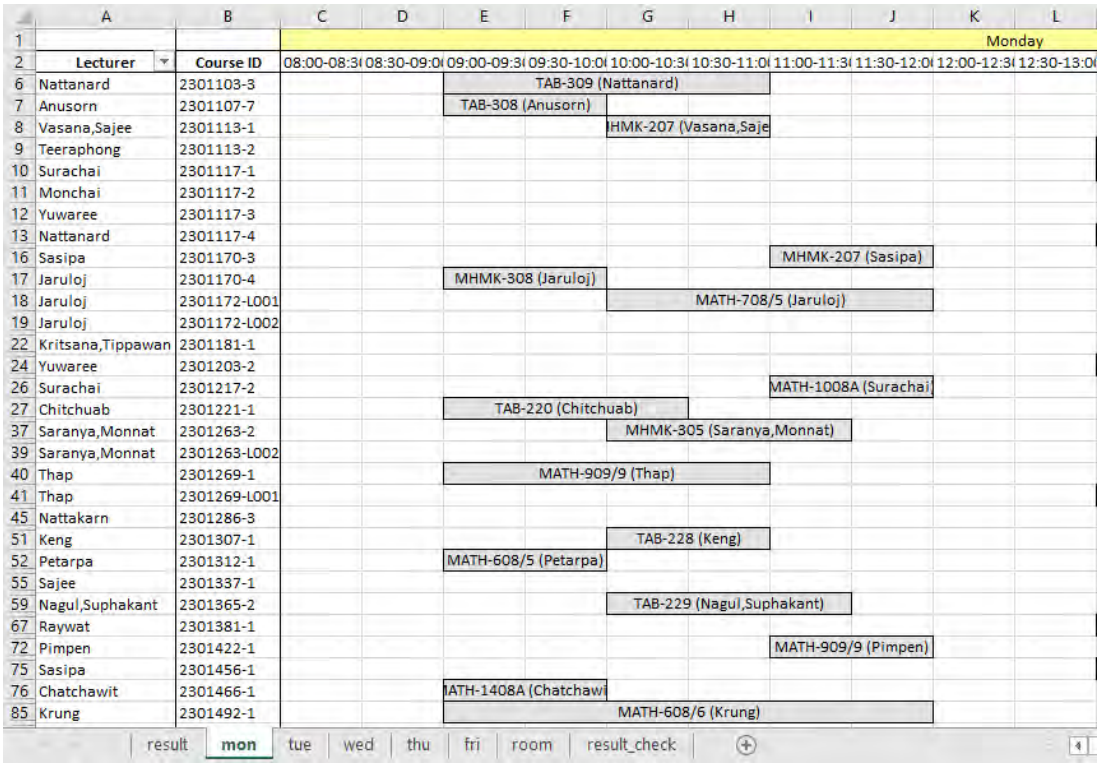


Figure 4.14: The final daily result in each day

In our result, teacher Krung and teacher Somjai have class meetings 12 time slots on Monday and Friday, respectively (see in Figures 4.15 and 4.16). In fact, we can reduce the maximum time slots from 12 time slots (in Constraint 8) into 10 time slots to reduce teaching hours per day of teacher Krung and teacher Somjai. In further study, we may reduce the maximum time slots to 8 time slots in order to reduce the teaching load of each lecturer by hiring new lecturers or expanding the prefer time slots so that the teaching load could spread to other lecturers/ times. In our model, we also set the maximum time slots per day of a student to be 12 (in Constraint 9). Since some students may have classes with a lecturer who teaches 12 time slots in a day. Therefore, the maximum time slots in both Constraints 8 and 9 need to be the same number.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1		1=assigned	Monday																	
2	**Lecturer**	**Class meeting**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
80	Krung	2301481-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
85	Krung	2301492-1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
106	Krung	2301694-1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0

Figure 4.15: Teacher Krung has to teach 12 time slots on Monday

	A	B	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN
1		1=assigned	Friday																	
2	**Lecturer**	**Class meeting**	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
34	Somjai	2301251-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
74	Somjai	2301451-1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
81	Somjai,Wutichai	2301490-1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0

Figure 4.16: Teacher Somjai has to teach 12 time slots on Friday

Chapter 5
Conclusion

5.1 Conclusion

This project studied classroom scheduling problem with integer linear program that uses the data of first-semester in 2019 of the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University as a case study. We adjusted data and set constraints to our model. The constraints are

1. Each class meeting must be assigned to some time slots which is equivalent Constraint 1 in Section 3.
2. The lecturers must not attend more than one class meeting at a time.
3. The students must not attend more than one class meeting at a time.
4. The lecturer must be available when the class meetings requiring.
5. The room must be available when the class meetings requiring.
6. All class meetings must be not assigned to unavailable time slots.
7. Either a class meeting c is assigned in time slot t or a class meeting c is not assigned in time slot t.
8. Maximum total time slots that a lecturer can teach in a day is 12.
9. Maximum total time slots that a student can attend in a day is 12.
10. Maximum consecutive time slots for a lecturer in a day is 8.

11. Maximum consecutive time slots for a student in a day is 8.
12. Each lecture of a class meeting may have consecutive time slots.
13. Class meetings should not have more than one lecture on the same day.
14. Class meetings should have a day break between 2 consecutive lectures.

We are successfully create an effective room timetable for the Department of Mathematics and Computer Science under our restrictions. We use CPLEX Studio IDE software version 12.6.3 and Python version 3.7 to solve this problem. The CPLEX takes time to process 2:18:94 minutes with 72 courses and 106 class meetings by notebook computer Intel(R) Core(TM) i7-4710HQ.

5.2 Suggestions

1. Use Access instead of Excel because Access is more convenient when changing the size of data.
2. Be more appropriate in dividing the room types. This may solve the problem of no available rooms with the seats more than number of students.
3. Should divide “Lab” room type as small lab, medium lab, and large lab.
4. Reduce the maximum time slots that a lecturer can teach in a day from 12 into 8 time slots by adding lecturers: teaching load can spread to other lecturers or adding preference time slots: teaching hours can spread to other days that teacher Krung or Somjai have fewer teaching time slots than 8 slots.

References

[1] A. Gunawan, K.M. Ng, and K.L. Poh. Solving the Teacher Assignment-Course Scheduling Problem by a Hybrid Algorithm. *World Academy of Science, Engineering and Technology*, Vol. 1, No. 9, (2007) : pp. 491-496.

[2] A. Oner, S. Ozcan, and D. Dengi. *Optimization Of University Course Scheduling Problem With A Hybrid Artificial Bee Colony Algorithm*. Department of Industrial Engineering, Yasar University, 2011.

[3] J. Gao. *Branch and bound (BB)* [Online]. (16 April 2014). Available from: https://optimization.mccormick.northwestern.edu/index.php/Branch_and_bound_(BB) [20 March 2020].

[4] M. Assi, B. Halawi, and R.A. Haraty. Genetic Algorithm Analysis using the Graph Coloring Method for Solving the University Timetable Problem. *Procedia Computer Science*, Vol. 126, (2018) : pp. 899-906.

[5] N.A.H. Aizam and L. Caccetta. Computational models for timetabling problem. *Numerical Algebra, Control and Optimization*, Vol. 4, No. 3, (September 2014) : pp. 269-285.

[6] P. Avella and I. Vasil'EV. A computational study of a cutting Plane algorithm for university course timetabling. *Journal of Scheduling*, Vol. 8, (2005) : pp. 497-514.

[7] P. Kostuch. *The University Course Timetabling Problem with a Three-Phase Approach*. Department of Statistics, Oxford University, 2005.

[8] P. Li. *Cutting Plane Method* [Online]. (20 January 2015). Available from: https://www.math.cuhk.edu.hk/course_builder/1415/math3220/L5.pdf[20 March 2020].

[9] P. Pisut. *กำหนดการเชิงเส้น (Linear Programming)*. 1^{st} ed. Bangkok: Danex Intercorporation, 2018.

[10] S. Abdullah, E.K. Burke, and B. McCollum. A Hybrid Evolutionary Approach to the University Course Timetabling Problem. *Congress on Evolutionary Computation*, (2007) : pp. 1764-1768.

[11] S. Daskalaki, T. Birbas, and E. Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, Vol. 153, (2004) : pp. 117-135.

[12] S. Ghaemi, M.T. Vakili, and A. Aghagolzadeh. *Using a Genetic Algorithm optimizer tool to solve University Timetable Scheduling Problem*. Faculty of Electrical and Computer Engineering, University Of Tabriz, 2007.

Appendix

Appendix A: The teaching timetable of the department

Figure A1: QR code of the teaching timetable of the department

URL: https://drive.google.com/open?id=1yjXlpXrRhNm_yxqSq2cda0bceYbtLlmN

Appendix B: The room timetable of the faculty

Figure A2: QR code of the room timetable of the faculty

URL: https://drive.google.com/open?id=1GxJWCchIQTZPMcAYMNXSAf7F9LnJtnIk

Appendix C: The modified timetable timetable of CPLEX

Figure A3: QR code of the modified timetable timetable of CPLEX

URL: https://drive.google.com/open?id=1Ll8W7yI4r3mZ3lSggtVc5zrnzI1UsOun

Appendix D: Code for CPLEX

Figure A4: QR code of the code for CPLEX

URL: https://drive.google.com/open?id=1yWbrdS_Jn8JKiSVo8MWs3XU6_Vokjtzh

Appendix E: Data for Python

Figure A5: QR code of the data for Python

URL: https://drive.google.com/open?id=1vHFdaHHBG-ECFP33Jbo1WpXkjYkAhxcm

Appendix F: Code for Python

Figure A6: QR code for the code for Python

URL: https://drive.google.com/open?id=1mtDZvMr2RaGaUsiK0F5A1fzQvMT-iyzG

Appendix G: Improved code for Python

Figure A7: QR code for the improved code for Python

URL: https://drive.google.com/open?id=1xD3IzVej3Akm8ak40FmmkPC3HkHJBApb

Appendix H: Result from Python

Figure A8: QR code for the result from Python

URL: https://drive.google.com/open?id=1OfWj7xyrDSCs-DU08FBWFSTYt2UiMZzY

Appendix I: Result from Python after improvement

Figure A9: QR code for the result from Python after improvement

URL: https://drive.google.com/open?id=1MI3eoKW0P9tR9aq88adodUuv3ChJibQK

Appendix J: Project Proposal

The Project Proposal of Course 2301399 Project Proposal

Academic Year 2019

Project Tittle (Thai)	การแก้ปัญหาการจัดตารางห้องเรียนด้วยวิธีการกำหนดการเชิงเส้นจำนวนเต็ม
Project Tittle (English)	Classroom Scheduling Problem with an integer linear programming method
Project Advisor	Associate Professor Phantipa Thipwiwatpotjana, Ph.D.
By	Miss Chanida Leelayutto ID 5933511023 Mathematics, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University

---

Background and Rationale

Classroom Scheduling Problem is very difficult to solve by hand. People normally replicate the timetables of previous years with few changes to accommodate recent situations to update the timetable but it maybe not effective.

This project will try to manage the restrictions such as the limited number of classrooms, the available lecturers / classrooms at each time period, and class meeting conflict of each lecturer / student (one class at a time). Then, we apply an integer linear programming method learned in the course of Operations Research II to find the optimal solution to this problem.

Objectives

Solve the Classroom Scheduling Problem by using CPLEX Studio IDE software version 12.6.3.

Scopes

1. We use the data from the first-semester of 2018 provided by the Department of Mathematics and Computer Science and the Office of the registrar.

2. We consider only courses in the Department of Mathematics and Computer Science and classroom in Maha Vajirunhis building, Mahamakut building, and Tab Nilaniti building.

3. We use CPLEX Studio IDE software version 12.6.3 to solve the Classroom Scheduling Problem.

Project Activities

1. Study the Classroom Scheduling Problem.

2. Combine related data and research papers.

3. Study the model for timetabling problem in [4].

4. Build an integer linear programming model for solving the problem.

5. Study how to use CPLEX Studio IDE software version 12.6.3.

6. Code the model on CPLEX Studio IDE software version 12.6.3.

7. Recheck and modify the model.

8. Conclude the results and write a report.

Duration

Procedue	Month								
	Aug.	Sep.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.
1.Study the Classroom Scheduling Problem.									
2.Combine related data and research papers.									
3.Study the model for timetabling problem in [4].									
4.Build an integer linear programming model for solving the problem.									
5.Study how to use CPLEX Studio IDE software version 12.6.3.									
6.Code the model on CPLEX Studio IDE software version 12.6.3.									
7.Recheck and modify the model.									
8.Conclude the results and write a report.									

Benefits

1. The benefits of the project owner.

    (a) Apply knowledge from the Operations Research II course solved the Classroom Scheduling Problem.

    (b) Use CPLEX Studio IDE software version 12.6.3 solved the Classroom Scheduling Problem.

2. The benefits of project users.

    (a) Increase convenient and effective to assign the Classroom Scheduling Problem.

    (b) Can use the integer linear programming method solved the Classroom Scheduling Problem and can apply to the other timetabling problem.

Equipments

1. Hardware

    (a) Notebook computer Intel(R) Core(TM) i7-4710HQ

    (b) Printer

    (c) Flash drive

2. Software

    (a) CPLEX Studio IDE software version 12.6.3

    (b) Excel 365

    (c) TeXstudio software version 2.12.16

Budgets

1.	Flash drive	140	Baht
2.	Laser Pointer	900	Baht
3.	Tray DVD Drive for HHD	280	Baht
4.	SSD 240 GB	920	Baht
5.	Notebook Service Fee	500	Baht
6.	Battery Notebook	1,260	Baht
7.	Binding the report	1,000	Baht
Total		5,000	Baht

Appendix

Parameters

Let C be a set of class meetings,

R be a set of room types available,

L be a set of lecturers,

S be a set of student groups,

T be a set of timeslots,

C_r be class meetings requiring a room of type r, $\forall r \in R$,

C_l be class meetings that are taught by lecturer l, $\forall l \in L$,

C_s be class meetings that have the same group student s, $\forall s \in S$,

N_r be the number of rooms of type r available, $\forall r \in R$, and

$p_{c,t}$ be the preference of having class meeting c at timeslot t.

Decision Variables

$$x_{c,t} = \begin{cases} 1 & \text{if a class of meeting } c \text{ is assigned in timeslot } t, \\ 0 & \text{Otherwise.} \end{cases}$$

Objective Function

The objective function is to maximize the total timeslot preference of assigning the class meetings:

$$\max \sum_{c} \sum_{t} p_{c,t} x_{c,t}.$$

Constraints

1. All class meetings must be assigned to a timeslot.
2. The number of classrooms is limited.
3. The lecturers must not teach more than one class meeting at a time.
4. The students must not study more than one class meeting at a time.

Integer linear programming formulation

Objective Function / Constraints	Mathematical language	OPL language
Maximize the total timeslot preference of assigning the class meetings.	$\max \sum_c \sum_t p_{c,t} x_{c,t}$	maximize sum (c in class, t in timeslot) p[c][t] * x[c][t];
All class meetings must be assigned to a timeslot.	$\sum_t x_{c,t} = 1, \quad \forall c \in C.$	forall (c in class) sum (t in timeslot) x[c][t] == 1;
The number of classrooms is limited.	$\sum_{c \in C_r} x_{c,t} \leq N_r, \quad \forall r \in R, \forall t \in T.$	forall (r in roomtype, t in timeslot) sum (c in classroom[r]: c in class) x[c][t] <= Nr[r];
The lecturers must not teach more than one class meeting at a time.	$\sum_{c \in C_l} x_{c,t} \leq 1, \quad \forall l \in L, \forall t \in T.$	forall (l in lecturer, t in timeslot) sum (c in classlecturer[l]: c in class) x[c][t] <= 1;
The students must not study more than one class meeting at a time.	$\sum_{c \in C_s} x_{c,t} \leq 1, \quad \forall s \in S, \forall t \in T.$	forall (s in student, t in timeslot) sum (c in classstudent[s]: c in class) x[c][t] <= 1;

Biography

Chanida Leelayutto ID 5933511023

Department of Mathematics and Computer Science,

Faculty of Science, Chulalongkorn University