Wageningen University - Department of Social Sciences

Operations Research and Logistics

MSc Thesis

A Mixed-Integer Linear Programming Model for University Course Timetabling Problems

Abstract
University course timetabling models vary depending on the regulations and characteristics of different institutions. The purpose of this thesis is to develop a mathematical model for the university course timetabling problems that are inspired by the real situation at Wageningen University. A systematic literature review is conducted first to summarize the objectives and constraints that are commonly used in literature. Then based on the author's experiences of studying at Wageningen University, a tailored university course timetabling problem is defined, which includes two objectives and 21 constraints. An integer programming model is developed based on the problem, with the objective of maximizing lecturers' preferences for timeslots and days while at the same time minimizing the space waste in the course-room assignments. The performance of this model is examined in FICO Xpress solver with three instances. With small instances, the model can get an optimal solution after a short time, while with large instances, it can get a good feasible solution within an acceptable time. The timetables of the instances satisfy all the hard constraints imposed on the model.

Key words: University course timetabling, integer programming, optimization, mathematical model

Word count: 9115

April 2023

Student	Yuan Yao
Registration number	1053376
MSc program	Master Food Quality Management
Specialisation	Quality and Food Logistics
Commissioner	Wageningen University
Supervisor(s)	Peter Kirst
Examiner/2nd supervisor	Frits Claassen
Thesis code	**ORL**-80436

WAGENINGEN
UNIVERSITY & RESEARCH

A Mixed-Integer Linear Programming Model for University Course Timetabling Problems

Master Thesis

Operations Research & Logistics Group
Wageningen University & Research (WUR)
Wageningen, The Netherlands

Student name: Yuan Yao
Student number: 1053376
Supervisor: Peter Kirst
Course code: ORL-80436
Date: March 12, 2023

Abstract

University course timetabling models vary depending on the regulations and characteristics of different institutions. The purpose of this thesis is to develop a mathematical model for the university course timetabling problems that are inspired by the real situation at Wageningen University. A systematic literature review is conducted first to summarize the objectives and constraints that are commonly used in literature. Then based on the author's experiences of studying at Wageningen University, a tailored university course timetabling problem is defined, which includes two objectives and 21 constraints. An integer programming model is developed based on the problem, with the objective of maximizing lecturers' preferences for timeslots and days while at the same time minimizing the space waste in the course-room assignments. The performance of this model is examined in FICO Xpress solver with three instances. With small instances, the model can get an optimal solution after a short time, while with large instances, it can get a good feasible solution within an acceptable time. The timetables of the instances satisfy all the hard constraints imposed on the model.

Key words: University course timetabling, integer programming, optimization, mathematical model

Chapter 1

Introduction

Educational timetabling is a difficult task that educational institutions must perform regularly. It involves assigning courses between students and lecturers to times and rooms while avoiding conflicts as much as possible. Depending on the type of institutions (school or university) and the type of events (course or exam), it mainly consists of three categories: school timetabling, university exam timetabling, and university course timetabling (Schaerf, 1999). Usually the task of developing timetables is fulfilled by reusing the timetables from previous years and making some modifications manually to deal with new requirements. However, it is time-consuming and less flexible compared to automated timetabling, especially when there is a large change in the requirements. For this reason, a lot of research has been conducted on automated timetabling over the last decades.

This thesis focuses on university course timetabling. It basically involves assigning courses taught by a lecturer and taken by a certain number of students to a certain number of rooms and timeslots, while many requirements must be satisfied. These requirements are the reflection of the rules and policies set up by universities as well as the needs and preferences of students and lecturers (Aziz & Aizam, 2018). Over the last decades, automated university course timetabling has been studied extensively. There have been standard formulations and benchmark instances of university course timetabling problems based on international timetabling competitions (Ceschia Gaspero & Schaerf, 2022). In addition to that, studies have been devoted to solving the university course timetabling problems in specific real-world cases (Chen et al., 2021; Bakir & Aksop, 2008; Algethami & Laesanklang, 2021).

Different methods have been developed to solve course timetabling problems, which can be mainly classified into three categories: operational research methods, heuristics methods, and hybrid methods (Chen et al., 2021; Babaei, Karimpour & Hadidi, 2015). This thesis focuses on developing an integer programming model, which falls under operational research methods. An integer programming model is a mathematical optimization model which includes integer variables, constraints, and an objective function (Mikkelsen, 2021). The application of integer programming can easily and flexibly convert the rules and requirements into mathematical equations (Daskalaki, Birbas & Housos, 2004). Especially, binary variables (variables that can only take the value 0 or 1) are often used to formulate "yes or no" decisions. Integer programming is one of the exact techniques that can provide an optimal solution and can prove the optimality of the generated solution (Fonseca et al., 2017). It is a conventional method, of which the earliest studies can be traced back to Lawrie (1969). In the past, integer programming methods could only solve timetabling problems with limited sizes. However, in recent years, due to the development in computer software and hardware, integer programming methods have again started being an acceptable approach for large timetabling problems (MirHassani & Habibi, 2013).

However, despite the rich literature on university course timetabling problems, due to the variations of policies and rules among universities, the timetabling model built for one university is not applicable to another one (Schimmelpfeng & Helber, 2007; Schaerf & Gaspero, 2007). Hence, the objective of this thesis is to build a university course timetabling model for master programs by considering the special rules and requirements at Wageningen University through 0-1 integer programming. To achieve the objective, three research questions are formulated:

1. Which objectives and constraints of the university course timetabling models in the literature are suitable for the course timetabling problems that are inspired by the real situation of master programs at Wageningen University?

2. What is a suitable integer programming model for the university course timetabling problems that are inspired by the real situation of master programs at Wageningen University?

3. What is the performance of the integer programming model for the university course timetabling problem that are inspired by the real situation of master programs at Wageningen University?

For the first research question, a literature review was conducted to have an overview of the objectives and constraints that other studies concerned. The aim is to ensure a comprehensive inclusion of possible requirements. Another aim is to have a deep insight into how other studies formulated mathematical equations through integer programming. For example, what kinds of variables they used, or how they formulated the objective function. Then, based on the author's experiences of studying at Wageningen University, the suitable objectives and constraints are selected from the review, and also some new requirements are added to the problem. Afterwards, they are transformed into the integer programming model. For the last question, the mathematical model will be tested in the optimization solver FICO Xpress to examine its performance.

This thesis is divided into six chapters. In Chapter 2, the relevant literature on solving university course timetabling problems through integer programming is reviewed. In Chapter 3, first a basic integer model is presented which only involves assigning courses to rooms. Then, a sophisticated model is built to solve the course timetabling problem inspired by the real situation of master programs at Wageningen University. In Chapter 4, the results of three instances with different sizes are presented after testing them in FICO Xpress to examine the performance of the model. Chapter 5 discusses the performance of the extended model, in terms of its advantages and limitations. Finally, Chapter 6 includes the main conclusions and limitations of this research, as well as some recommendations for future research.

Chapter 2

Literature review

A systematic literature review is conducted to answer the first research question. The review is based on the review article by Abidi, Leeuw & Klumpp (2014). There are 3 steps, searching, screening, and reporting. These processes are explained in detail in Appendix A. Finally, ten articles are selected for the systematic review. The summarized results are presented below. This chapter is divided into two sections. In the first part, the objectives and constraints used in the ten articles are summarized. The second part focuses on the methods in the literature about how to formulate the decision variables and objective functions.

2.1 Objectives and constraints of university course timetabling

According to the literature, there are five basic elements of a university course timetable, which are courses, rooms, time, lecturers, and students (Daskalaki, Birbas & Housos, 2004; Aziz & Aizam, 2018). Among them, courses, rooms, and time are the main components of a timetable. The complexity of the timetabling problem increases with the elements involved. To make a more clear description, we start with the definitions of some terms:

1. Lecture: a scheduled activity between teachers and students.
2. Course: a course consists of a set of lectures.
3. Timeslot: a time interval during which a lecture is scheduled, such as 8 - 10 a.m. on a certain day.

To have a better understanding, at first, the process of how to solve university timetabling problems through integer programming is described. First, the rules and policies of the university are listed, as well as the needs of the stakeholders involved in timetabling, such as lecturers and students. These requirements are classified into two categories: hard constraints and soft constraints (Babaei, Karimpour & Hadidi, 2015). Hard constraints cannot be violated to deliver a feasible solution. For instance, a lecturer cannot give more than one lecture at a specific timeslot. Soft constraints shall be satisfied as much as possible to ensure the quality of a timetable, such as lecturers' preferences (MirHassani & Habibi, 2013). Then, the hard constraints of a timetabling problem are transformed into the constraints in the mathematical integer programming model, while the soft constraints are usually embedded in the objective function in that model (MirHassani & Habibi, 2013). After the formulation of the complete model, it will be validated in optimization solvers with either real-world data or simulated data. The general steps are summarized in Table 2.1.

Table 2.1: Steps to solve a university course timetabling problem through integer programming

Step 1	list the rules and requirements for the timetable
Step 2	identify the hard and soft constraints
Step 3	convert the hard and soft constraints into mathematical model
Step 4	validate the model in optimization solvers

Though the constraints vary in different articles, there are important types of hard constraints that can never be violated, otherwise, the timetable would be infeasible (Daskalaki, Birbas & Housos, 2004). Such constraints will always be present in models, which refer to:

1. Avoiding conflicts in a timetable. A conflict occurs when, for example, more than one lecture is scheduled in the same room at the same time; or when more than one lecture is assigned to a lecturer or student at the same time.

2. Completeness of a timetable. All the courses shall be scheduled at right time, in the right room.

In addition to the constraints above, Table 2.2 below shows the requirements used by the ten reviewed articles. Considering the limited space, these articles are denoted by numbers ([1], [2], ..., [10]) here in order to put them on one page. They can be found by the corresponding numbers in Bibliography. We can see that the most frequently used objective is to optimize lecturers' or students' preferences (R1). Besides R1, the most frequently used constraints are R2, R5, R6, R7, and R13, which appeared in at least five research.

It's worth mentioning that some researchers consider certain constraints in the table as hard constraints, while others consider them as soft constraints, depending on their research aims. For example, articles [4], [8], and [9] considered R6 (spreading lectures over the week) as a soft constraint, while articles [1], [2], [7], and [10] regarded it as a hard one. The same situation also happens to R3 (room stability) and R7 (workload of lecturers).

Table 2.2: Requirements of course timetabling problems

NO.	Requirements	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
R1	Maximizing the preferences of lecturers or students	-	O	O	-	-	O	O	-	O	O
R2	Room capacity	H	H	-	H	-	-	H	-	H	H
R3	Room stability (scheduling all the lectures of a course in the same room)	-	-	H	O	-	-	H	-	-	-
R4	Precedence of lectures or courses	H	-	H	-	-	-	H	-	-	H
R5	Avoiding courses in certain timeslots	H	H	H	-	-	-	-	H	-	H
R6	Lectures of a course spread out the week	H	H	-	O	-	-	H	O	O	H
R7	Workload of lecturers	O	H	-	-	H	O	-	H	-	H
R8	Compactness of timetable	-	-	-	O	-	-	-	-	-	-
R9	Availability of lecturers and rooms at a certain timeslot	-	H	-	H	-	H	-	-	-	H
R10	Avoiding scheduling some courses on the same day	-	-	-	-	-	-	-	-	-	H
R11	Pre-assigned courses	-	-	H	-	H	-	-	-	-	-
R12	Minimizing the cost of assigning courses to different time periods	-	-	-	-	O	-	-	-	-	-
R13	Requirements on specific equipment or room features	H	H	-	H	-	-	H	-	H	-
R14	Consecutive timeslots for courses that cover more than one timeslot	H	-	H	-	H	H	-	-	-	-
R15	Lecturers' expertise to teach a specific course	H	H	-	-	-	-	-	H	-	-
R16	The number of days a student has to visit the campus	-	O	-	-	-	-	-	-	-	-
R17	Minimizing the difference between the room capacity and the number of students of the course	-	-	-	O	-	-	-	-	-	-

H: requirements deemed hard constraints by the research

O: requirements deemed soft constraints by the research and included in the objective function

-: not applicable

2.2 Formulation of decision variables and objective functions

According to ten reviewed articles, there are usually two types of binary variables, which are basic variables and auxiliary variables (Daskalaki, Birbas & Housos, 2004). A set of basic variables have several indexes like course, room, time, lecturer, and student group. They are the main decision variables and take the burden of assigning rooms, time and lecturers to courses, and form most of the constraints. They take the value 1 to denote that a course is scheduled in a certain room, at a certain time, by a certain lecturer and to a certain student group. Besides, auxiliary variables assist to simplify the formulation of constraints. They are additional decision variables introduced to represent intermediate values or relationships in the model. For example, Havås et al. (2013) used 0-1 auxiliary variables to enforce the requirement of room stability.

There are several ways to develop objective functions. For example, Zaulir, Aziz & Aizam (2022) maximized teachers' preferences for time and rooms by assigning a value to each decision variable to denote the degree of preference. Higher values represent higher preferences. Another way is giving a penalty value to each decision variable to denote the degree of dislike, and the objective is to minimize the sum value of dislike. For example, according to Havås et al. (2013), a high value was given to the variables where the day is less preferred, while a low weight was given to those with the day preferred by the students. For multi-objective functions, in the model of Algethami & Laesanklang (2021), different goals were simply added together in the objective function. Colajanni (2019) assigned a penalty weight to each goal and the objective is to minimize the sum value of all the goals. A larger penalty weight denotes higher importance.

Chapter 3

Methodology

This chapter is partitioned into three sections. In Section 3.1, a very simple course timetabling model is introduced, which only involves course-room assignments. It can help readers to understand the extended timetabling model better. In Section 3.2, the university course timetabling problem is demarcated based on results found in the literature review and the author's experience of studying at Wageningen University. This section aims to describe a detailed and complete university course timetabling problem. In Section 3.3, a sophisticated course timetabling model is developed based on the problem.

3.1 Example of a basic university course timetabling model

A basic problem is described as follows. A set of courses need to be scheduled in a certain number of available rooms. To avoid clashes, a room cannot be allocated to two courses, while a course can only be given in exactly one room. Besides, the room allocated to the course shall be large enough to hold all the students of that course.

The set of rooms is denoted by R, and the set of courses is denoted by C. Parameter C_r denotes the capacity of room r, which means how many students the room can hold. Parameter N_c denotes the number of students of course c. Parameter $P_{c,r}$ denotes the lecturers' preferences for course c and room r. Higher values represent higher preferences. The sets and parameters are displayed in Table 3.1.

Decision variable $x_{c,r}$ is a binary variable. It takes the value 1 if course c is scheduled in room r, while it takes the value 0 if course c is not scheduled in room r.

Table 3.1: Definition of sets and parameters

Sets	Definition
C	set of courses
R	set of rooms
Parameters	**Definition**
C_r	the capacity of room r
N_c	the number of students of course c
$P_{c,r}$	the lecturer's preferences for course c and room r

The objective function is to maximize the total preferences of the lecturers for course c and room r. It is expressed as:

$$\max \quad \sum_{c \in C} \sum_{r \in R} P_{c,r} \cdot x_{c,r} \tag{3.1}$$

There are three constraints. The mathematical equations are shown below. First, equation (3.2) ensures that each room r can hold at most one course c. Second, equation (3.3) represents that every course c needs to be assigned to exactly one room r. Last, equation (3.4) ensures that room r should be large enough for course c.

$$\sum_{c \in C} x_{c,r} \leq 1, \qquad \forall r \in R \tag{3.2}$$

$$\sum_{r \in R} x_{c,r} = 1, \qquad \forall c \in C \tag{3.3}$$

$$N_c \leq C_r + (1 - x_{c,r}) \cdot N_c, \qquad \forall c \in C, \quad \forall r \in R \tag{3.4}$$

$$x_{c,r} \in \{0,1\}, \qquad \forall c \in C, \forall r \in R \tag{3.5}$$

3.2 Problem demarcation and description

Before the formulation of an extended mathematical model, a detailed and complete university course timetabling problem is demarcated in order to establish a model that is suited to a situation similar to Wageningen University & Research. This problem combined some of the constraints summarized in the literature review and some new constraints based on the author's experience of studying at Wageningen University. Then, according to the demarcation, a detailed description of the university course timetabling problem is stated.

First, some constraints summarized from the literature review are excluded. Table 3.2 shows the inclusion and exclusion of those constraints summarized in Table 2.2. An explanation is given for each excluded constraint. The included constraints are identified as hard or soft constraints. The soft constraints are then embedded in the objective function. We can see R8, R9, R11, R12, R14 and R16 will not be included in this thesis.

Besides, four hard constraints are added to the problem to make it more complete and specific.

1. Lecturer-course consistency. According to the author's experience, the lectures of some courses are given by different lecturers. In this problem, to simplify the situation, it assumes that all the lectures of a course will be taught by only one lecturer.

2. Time-course consistency. All the lectures of a course shall be scheduled at the same timeslot on different days (e.g. either in the morning or in the afternoon), just like what the author perceived at Wageningen University.

3. Courses in consecutive periods. According to the author's experience, some courses have to be scheduled in consecutive periods as they form a complete project.

4. According to the university regulations, there is a certain number of courses in each period for each master program.

The model aims to help create a timetable that contains the weekly schedule of courses in different periods for different student groups. Based on the demarcation, the complete university course timetabling problem is described as follows. A set of courses are given by a set of lecturers and taken by students from different master programs. They need to be scheduled in suitable rooms, at certain timeslots, days, and periods. A course has a certain number of lectures per week. A day is divided into several timeslots. A lecture covers only one timeslot. Table 3.3 introduces the objectives and constraints of the university course timetabling problem in detail.

Table 3.2: Inclusion and exclusion of requirements

Requirements	Identified as	Explanation
R1: the time preferences of lecturers or students	O	
R2: room capacity	H	
R3: room stability	H	
R4: precedence of lectures or courses	H	
R5: avoiding courses in certain timeslots	H	
R6: lectures of a course spread out over the week	H	
R7: workload of lecturers	H	
R8: compactness of timetable	-	Compactness of a timetable is measured for students who need to take multiple courses in a day. It aims to reduce "isolated lectures" that do not have adjacent lectures. It is not necessary here as there are only two courses per day at Wageningen University.
R9: unavailability of specific lecturers and rooms	-	This is a common constraint in literature. However, to simplify the model, it is assumed that lecturers and rooms are available at any timeslots.
R10: some courses shall not be scheduled on the same day	H	
R11: pre-assigned courses	-	This constraint will be excluded as there is no information about it based on the author's experience.
R12: the cost of assigning courses to different time periods	-	It is not a frequently used objective in literature, so it will not be included.
R13: courses require specific equipment or room features	H	
R14: consecutive timeslots for courses that cover more than one timeslot	-	This is suitable for the cases when timeslot is set as a certain number of minutes (e.g. 5 minutes) and a lecture may cover several consecutive timeslots. It is not applicable here as it assumes a lecture covers only one timeslot.
R15: lecturers' ability to teach a specific course	H	
R16: the number of days a student has to visit the campus	-	At Wageningen University, usually students need to visit the campus every weekday.
R17: minimize the difference between the room capacity and the number of students of the course	O	
H: requirements deemed hard constraints / O: requirements deemed soft constraints and included in the objective function / -: not applicable		

Table 3.3: Objectives and constraints of the university course timetabling problem

	Objectives
O1	Maximizing lecturers' preferences for weekdays and timeslots.
O2	Minimizing the difference between the number of seats in a room and the number of students of the course assigned to that room. A room shall be able to contain all the students of the course, but at the same time, it is suggested to assign a course with a student number close to the capacity of that room to avoid wasting a large room on a course with a small student group.
	Constraints
C1	All the lectures of a course are assigned.
	Course-room constraints:
C2	A room can hold at most one course at a certain timeslot.
C3	Each lecture shall be assigned exactly one room.
C4	The lectures of a course shall be scheduled in the same room (room stability).
C5	A room must have enough seats for all the students of that course.
C6	Some courses require specific room features or equipment.
	Course-lecturer constraints:
C7	A lecturer can teach at most one course at a time.
C8	Each lecture shall be taught by exactly one lecturer.
C9	The lectures of a course shall be taught by the same lecturer.
C10	A course can only be taught by a lecturer who has the relevant expertise.
C11	The number of courses taught by a lecturer shall not exceed the maximum workload.
	Course-time constraints:
C12	To avoid conflicts within a student group, there shall be at most one course at a time for each student group.
C13	According to the university regulations, there is a certain number of courses in each period for each master program.
C14	The lectures of a course are scheduled at the same timeslot on different days.
C15	No courses are scheduled at certain timeslots in certain periods according to the arrangements of the university.
	Spreading lectures of a course over the week:
C16	There is no more than one lecture per day for each course.
C17	When there are only two lectures per week for a course, there shall be at least one day off between the two lectures. There shall be no lecture on Friday.
C18	When there are three lectures per week for a course, they shall not be scheduled consecutively. On Thursday and Friday, there shall be at most one lecture.
	Course setups:
C19	Some courses shall not be scheduled in the same period.
C20	Some courses have to be scheduled in consecutive periods.
C21	Some elementary courses are the prerequisites for other courses, so they shall be scheduled before other courses.

3.3 Extended model formulation

According to the problem described in the previous section, this section introduces the definition of sets and parameters, the decision variables, the objective function, and the constraints of the extended model.

3.3.1 Definition of sets and parameters

First, there are five elements involved in the model, which are rooms, students, courses, lecturers, and time. These are introduced below respectively. The definitions of the sets and parameters of this extended model are displayed in Table 3.4 below.

Rooms

There are a certain number of rooms available with certain capacities. Let R denote rooms, and the capacity of room r is denoted by C_r. Besides, a course may need a room with certain features, such as equipment or facilities. Let Φ denote the set of features of rooms. Binary parameter $K_{r,\phi}$ denotes the room-feature matrix, where it takes a value 1 if room r has feature ϕ.

Students

A master program has more than one specialization and the courses vary depending on the specialization. Considering this, instead of dividing students based on their programs, students taking the same courses can be classified into the same group. Let S denote the set of student groups, i.e. $s \in \{1, 2, ..., S\}$.

Courses

A certain number of courses need to be scheduled. Let C denote the set of courses. The courses are divided into subsets, and each subset contains the courses for one group of students s. Therefore, the subset of the courses for each student group is denoted by C_S. For example, the set of courses for student group 1 is denoted as C_1. Each course c has a certain number of students, which is denoted by N_c. Each course has several lectures per week, which is denoted by Λ_c. Besides, let $U_{c,\phi}$ denote the course-feature matrix, where it takes the value 1 if course c requires room feature ϕ.

Lecturers

A certain number of lecturers need to be assigned to courses. It assumes that all the lectures of one course are given by the same lecturer. Lecturers are assumed to be available at any timeslots. Let L denote the set of lecturers. Let binary parameter $A_{c,l}$ denotes the course-lecturer matrix, where it takes the value 1 if lecturer l is able to teach course c. Besides, a maximum workload is set for lecturers, which is denoted by M_l.

Time

This model aims to create a timetable that schedules the weekly lectures of a course. It means the schedule for a course remains the same every week. It can schedule each lecture in a week for a course at a specific timeslot, on a specific weekday, and in a specific period (semester). Therefore, there are three types of time elements, which are timeslots on a day, weekdays, and periods. Let T denote the set of timeslots on a day. The total number of timeslots per day is represented by N_t. Let D denote the set of weekdays, from Monday to Friday, i.e. $d \in \{1,2,3,4,5\}$. Finally, let P denote the set of periods. The total number of periods involved in the timetabling problem is represented by N_p. Besides, according to university regulations, a certain number of courses shall be scheduled for each student group in a period (C11 in Table 3.3). The number of courses in period p for each student group is denoted by $\widetilde{C}_p$. Also, no lectures are scheduled at certain timeslots in certain periods (C13 in Table 3.3). Let $\widehat{T}_{t,p}$ denote the pair set of timeslot and period when no lectures are scheduled.

3.3.2 Decision variables

In the extended model, five sets of binary variables are required, which are:

- Basic variable $x_{c,r,l,d,t,p}$, which takes the value 1 when course c is given in room r, by lecturer l, on day d, at timeslot t, and in period p. Otherwise it is 0.
- Auxiliary variable $y_{c,r}$, which takes the value 1 when course c is scheduled in room r. Otherwise it is 0.
- Auxiliary variable $z_{c,l}$, which takes the values 1 when course c is taught by lecturer l. Otherwise it is 0.
- Auxiliary variable $\alpha_{c,t,p}$, which takes the value 1 when course c is scheduled at timeslot t in period p. Otherwise it is 0.
- Auxiliary variable $\beta_{c,p}$, which takes the value 1 when course c is scheduled in period p. Otherwise it is 0.

Table 3.4: Definition of sets and parameters

Sets	Definition
R	set of rooms, $r \in R$
C	set of courses, $c \in C$
S	set of student groups, $s \in S$
C_S	set of courses for different student groups, $C_1 \cup C_2 ... \cup C_S = C$
D	set of weekdays, from Monday to Friday, $d \in \{1,2,3,4,5\}$
T	set of timeslots, $t \in T$
P	set of periods, $p \in P$
L	set of lecturers, $l \in L$
Φ	set of room features, $\phi \in \Phi$
H	set of courses in pairs (c_m, c_n) that shall not be assigned in the same period, $\forall (c_m, c_n) \in C$
O	set of courses in pairs (c_m, c_n) that shall be assigned consecutively over the s periods, $\forall (c_m, c_n) \in C$
Q	set of courses in pairs (c_m, c_n) that c_m shall be assigned to the period before c_n, $\forall (c_m, c_n) \in C$
V	set of courses that have two lectures per week, $c_v \in C$
G	set of courses that have three lectures per week, $c_g \in C$
$\widehat{T}_{t,p}$	pair set of timeslot and period (t, p) when no courses shall be scheduled at timeslot t in period p
Parameters	**Definition**
C_r	the capacity of room r
N_c	the number of students that attend the course c
Λ_c	the number of lectures per week of course c
$K_{r,\phi}$	0-1 matrix; $K_{r,\phi} = 1$ if room r has feature ϕ
$U_{c,\phi}$	0-1 matrix; $U_{c,\phi} = 1$ if course c requires feature ϕ
$\widetilde{P}1_{l,d}$	the preferences of lecturer l to work on weekday d
$\widetilde{P}2_{l,t}$	the preferences of lecturer l to work at timeslot t
M_l	the maximum workload for lecturer l
$A_{c,l}$	0-1 matrix; $A_{c,l} = 1$ if lecturer l is able to teach course c
N_p	the number of periods involved in the course timetabling problem
N_t	the number of timeslots per weekday involved in the course timetabling problem
$\widetilde{C}_p$	the number of courses in period p for each student group
w_1	the weight assigned to the goal of lecturers' preferences
w_2	the weight assigned to the goal of difference between room capacity C_r and student number N_c

3.3.3 Objective function

The objective function consists of two goals as listed in Table 3.3. The weighted sum method is used to balance the two goals. In addition, to avoid the influence caused by the different units and scales of both goals, the two goals are normalized here to make the objective function unitless. The first goal is to maximize the total preferences of lecturers for weekdays and timeslots (O1). Coefficients $\widetilde{P1}_{l,d}$ and $\widetilde{P2}_{l,t}$ represent the preferences of lecturer l for working on day d and timeslot t respectively. Higher coefficients denote higher preferences. It is written as:

$$\max \quad \left\{ \sum_{c\in C}\sum_{r\in R}\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} (\widetilde{P1}_{l,d} + \widetilde{P2}_{l,t}) \cdot x_{c,r,l,d,t,p} \right\} \tag{3.6}$$

The second goal is to minimize the differences between the room capacity C_r and the student number of the course N_c (O2). It is written as:

$$\min \quad \left\{ \sum_{c\in C}\sum_{r\in R}\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} |C_r - N_c| \cdot x_{c,r,l,d,t,p} \right\} \tag{3.7}$$

To make the two goals consistent, the negative value of this function is taken. It is then a max function, which is written as:

$$\max \quad \left\{ -\sum_{c\in C}\sum_{r\in R}\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} |C_r - N_c| \cdot x_{c,r,l,d,t,p} \right\} \tag{3.8}$$

Let weight w_1 and w_2 denote the weight for each goal. The sum value of weight w_1 and w_2 equals 1. Therefore, the weighted objective function is written as:

$$\begin{aligned} \max \quad \{ & w_1 \cdot \sum_{c\in C}\sum_{r\in R}\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} (\widetilde{P1}_{l,d} + \widetilde{P2}_{l,t}) \cdot x_{c,r,l,d,t,p} \\ & + w_2 \cdot \sum_{c\in C}\sum_{r\in R}\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} (-|C_r - N_c| \cdot x_{c,r,l,d,t,p}) \} \end{aligned} \tag{3.9}$$

Let f_1 denote function (3.6) and f_2 denote function (3.8). The objective function (3.9) is denoted as W and is expressed as:

$$\max \quad \{ W = w_1 \cdot f_1 + w_2 \cdot f_2 \} \tag{3.10}$$

Then, the linear normalization method is used, which scales each value between 0 and 1 (Anagnostopoulos & Mamanis, 2010). The extreme values (maximum and minimum) of both goals need to be calculated. Let f_1^{max} and f_1^{min} denote the maximum and minimum value of f_1 and f_2^{max} and f_2^{min} denote the maximum and minimum value of f_2. The normalized f_1 and f_2 are denoted by μ_1 and μ_2, which are written as:

$$\mu_1 = \frac{f_1 - f_1^{min}}{f_1^{max} - f_1^{min}} \tag{3.11}$$

$$\mu_2 = \frac{f_2 - f_2^{min}}{f_2^{max} - f_2^{min}} \tag{3.12}$$

Finally, the objective function (3.10) can be normalized as:

$$\max \quad \{W = w_1 \cdot \mu_1 + w_2 \cdot \mu_2\} \tag{3.13}$$

3.3.4 Constraints

In this extended model, 21 constraints (from C1 to C21) summarized in Table 3.3 are considered. The mathematical equations are listed below. The first constraint ensures that for each course, all the lectures in a week are assigned (C1). It is enforced by equation (3.14). Here Λ_c represents the number of lectures per week for each course.

$$\sum_{r\in R}\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} x_{c,r,l,d,t,p} = \Lambda_c, \qquad \forall c \in C \tag{3.14}$$

Course-room constraints

To avoid clashes during assigning rooms to courses, a room can hold at most one course at a certain time (C2). This is enforced by equation (3.15). Each lecture shall be assigned to exactly one room (C3). It can be enforced by equation (3.16). This equation also ensures that the lectures of a course shall be scheduled in the same room (C4). Here binary variable $y_{c,r}$ takes the value 1, if course c is scheduled in room r. By multiplying $y_{c,r}$ and Λ_c, all the lectures of course c in a week are scheduled in the same room r. Equation (3.17) ensures that a room is large enough for a course (C5). Equation (3.18) ensures that a course can only be assigned in a room which has the required room features or equipment (C6). Binary parameter $K_{r,\phi}$ takes the value 1 if room r has room feature ϕ, while binary parameter $U_{c,\phi}$ takes the value 1 if course c requires room feature ϕ.

$$\sum_{c\in C}\sum_{l\in L} x_{c,r,l,d,t,p} \leq 1, \qquad \forall r \in R, \forall d \in D, \forall t \in T, \forall p \in P \tag{3.15}$$

$$\sum_{l\in L}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} x_{c,r,l,d,t,p} = y_{c,r}\cdot\Lambda_c, \qquad \forall c\in C, \forall r\in R \tag{3.16}$$

$$\begin{aligned} &N_c \le C_r + (1 - x_{c,r,l,d,t,p})\cdot N_c,\\ &\forall c\in C, \forall r\in R, \forall l\in L, \forall d\in D, \forall t\in T, \forall p\in P \end{aligned} \tag{3.17}$$

$$\begin{aligned} &U_{c,\phi}\cdot x_{c,r,l,d,t,p} \le K_{r,\phi}\cdot x_{c,r,l,d,t,p},\\ &\forall c\in C, \forall r\in R, \forall l\in L, \forall d\in D, \forall t\in T, \forall p\in P, \forall\phi\in\Phi \end{aligned} \tag{3.18}$$

Course-lecturer constraints

First, a lecturer can teach at most one course at a time (C7). This is enforced by equation (3.19). Meanwhile, each lecture shall be taught by exactly one lecturer (C8). It can be enforced by equation (3.20). This equation also ensures that the lectures of a course are given by the same lecturer (C9). Here binary variable $z_{c,l}$ takes the value 1, if course c is taught by lecturer l. By multiplying $z_{c,l}$ and Λ_c, all the lectures of course c are taught by the same lecturer l. Besides, a course can only be taught by a lecturer who has the expertise (C10). It is enforced by equation (3.21). Binary parameter $A_{c,l}$ takes the value 1 if lecturer l is able to teach course c. In addition, equation (3.22) the number of courses taught by a lecturer shall not exceed the maximum workload M_l (C11).

$$\sum_{c\in C}\sum_{r\in R} x_{c,r,l,d,t,p} \le 1, \qquad \forall l\in L, \forall d\in D, \forall t\in T, \forall p\in P \tag{3.19}$$

$$\sum_{r\in R}\sum_{d\in D}\sum_{t\in T}\sum_{p\in P} x_{c,r,l,d,t,p} = z_{c,l}\cdot\Lambda_c, \qquad \forall c\in C, \forall l\in L \tag{3.20}$$

$$x_{c,r,l,d,t,p} \le A_{c,l}, \qquad \forall c\in C, \forall r\in R, \forall l\in L, \forall d\in D, \forall t\in T, \forall p\in P \tag{3.21}$$

$$1 \le \sum_{c\in C} z_{c,l} \le M_l, \qquad \forall l\in L \tag{3.22}$$

Course-time constraints

First, to avoid conflicts within each student group, at most one course can be scheduled at a certain time (C12). This is enforced by equation(3.23). According to the university regulations, a certain number of courses $\widetilde{C}_p$ shall be assigned to each student group s in period p (C13). This is reflected in equation (3.24). Meanwhile, the lectures of a course shall be scheduled at the same timeslot on different days (C14). This is enforced by equation (3.25). Here binary variable $\alpha_{c,t,p}$ takes the value 1, if course c is scheduled at timeslot t in period p. By multiplying $\alpha_{c,t,p}$ and Λ_c, all the lectures of course c on different days are scheduled at the same timeslot in the same period. Besides, according to the arrangement of the university, at certain timeslots, no courses are scheduled (C15). This is enforced by equation (3.26). Here (t, p) represents the pair set of timeslot and period when no courses are scheduled.

$$\sum_{c \in C_S} \alpha_{c,t,p} \leq 1, \qquad \forall t \in T, \forall p \in P \tag{3.23}$$

$$\sum_{c \in C_S} \beta_{c,p} = \widetilde{C}_p, \qquad \forall p \in P \tag{3.24}$$

$$\sum_{r \in R} \sum_{l \in L} \sum_{d \in D} x_{c,r,l,d,t,p} = \alpha_{c,t,p} \cdot \Lambda_c, \qquad \forall c \in C, \forall t \in T, \forall p \in P \tag{3.25}$$

$$\sum_{c \in C_S} \alpha_{c,t,p} = 0, \qquad \forall (t,p) \in \widehat{T}_{t,p} \tag{3.26}$$

Spreading lectures over the week

Besides the main characteristics, there are also some detailed requirements of this timetabling problem. First, to spread the lectures of a course over the week, there is no more than one lecture per day for each course (C16). This is reflected in equation (3.27). If a course has two lectures $\Lambda_c = 2$ per week, there should be at least one day off between the two lectures, and no lectures shall be scheduled on Friday (C177). This is reflected in equation (3.28) and (3.29). Similarly, if a course has three lectures $\Lambda_c = 3$ per week, the three lectures shall not be consecutive, and there shall be at most one lecture on Thursday and Friday (C18). This is covered in equation (3.30) and (3.31).

$$\sum_{r \in R} \sum_{l \in L} \sum_{t \in T} \sum_{p \in P} x_{c,r,l,d,t,p} \leq 1, \qquad \forall c \in C, \forall d \in D \tag{3.27}$$

$$\begin{aligned} & x_{c,r,l,d,t,p} + x_{c,r,l,d+1,t,p} \leq 1, \\ & \forall c \in V, \forall r \in R, \forall l \in L, \forall t \in T, \forall p \in P, \\ & \forall d \in \{1,2,3\} \end{aligned} \tag{3.28}$$

$$\begin{aligned} & x_{c,r,l,d,t,p} = 0, \\ & \forall c \in V, \forall r \in R, \forall l \in L, \forall d = \{5\}, \forall t \in T, \forall p \in P \end{aligned} \tag{3.29}$$

$$x_{c,r,l,d,t,p} + x_{c,r,l,d+1,t,p} + x_{c,r,l,d+2,t,p} \leq 2, \quad \forall c \in G, \forall r \in R, \forall l \in L, \forall t \in T, \forall p \in P \quad \forall d \in \{1,2,3\} \tag{3.30}$$

$$x_{c,r,l,d,t,p} + x_{c,r,l,d+1,t,p} \leq 1, \quad \forall c \in G, \forall r \in R, \forall l \in L, \quad \forall d = \{4\}, \forall t \in T, \forall p \in P \tag{3.31}$$

Course setup constraints

There are some requirements for the setup of courses. First, some courses c_m and c_n shall not be scheduled in the same period. It is reflected in equation (3.32) (C19).

$$(\beta_{c_m,p} + \beta_{c_n,p}) \leq 1, \qquad \forall (c_m, c_n) \in H, \forall p \in P \tag{3.32}$$

Second, some courses c_m and c_n shall be scheduled in the consecutive periods, which are reflected in equation (3.33), (3.34), (3.35) (C20).

$$\beta_{c_m,p} - \beta_{c_n,p+1} = 0, \qquad \forall (c_m, c_n) \in O, \forall p \in \{1,2,...,P-1\} \tag{3.33}$$

$$\beta_{c_m,p} = 0, \qquad \forall c_m \in O, p = \{P\} \tag{3.34}$$

$$\beta_{c_n,p} = 0, \qquad \forall c_n \in O, p = \{1\} \tag{3.35}$$

Third, some courses c_m, known as prerequisites, shall be scheduled before other courses c_n, as reflected in equation (3.36), (3.37), (3.38), (3.39) (C21).

$$\beta_{c_m,p} \leq \sum_{p=p+1} \beta_{c_n,p}, \qquad \forall (c_m, c_n) \in Q, \forall p \in \{1,2,...,P-2\} \tag{3.36}$$

$$\beta_{c_m,p} = \beta_{c_n,p+1}, \qquad \forall (c_m, c_n) \in Q, \forall p = \{P-1\} \tag{3.37}$$

$$\beta_{c_m,p} = 0, \qquad \forall c_m \in Q, \forall p = \{P\} \tag{3.38}$$

$$\beta_{c_n,p} = 0, \qquad \forall c_n \in Q, \forall p = \{1\} \tag{3.39}$$

Finally, equation (3.40) denotes the relationship between the two types of variables. Equation (3.41), (3.42), (3.43), (3.44), (3.45) enforce the five sets of decision variables to be binary.

$$\sum_{t \in T} \alpha_{c,t,p} = \beta_{c,p}, \qquad \forall c \in C, p \in P \tag{3.40}$$

$$x_{c,r,l,d,t,p} \in \{0,1\}, \qquad \forall c \in C, \forall r \in R, \forall l \in L, \forall d \in D, \forall t \in T, \forall p \in P \tag{3.41}$$

$$y_{c,r} \in \{0,1\}, \qquad \forall c \in C, \forall r \in R \tag{3.42}$$

$$z_{c,l} \in \{0,1\}, \qquad \forall c \in C, \forall l \in L \tag{3.43}$$

$$\alpha_{c,t,p} \in \{0,1\}, \qquad \forall c \in C, \forall t \in T, \forall p \in P \tag{3.44}$$

$$\beta_{c,p} \in \{0,1\}, \qquad \forall c \in C, \forall p \in P \tag{3.45}$$

Chapter 4

Results

This chapter introduces the research design and the computation results of three instances. In Section 4.1, first the optimization solver used is introduced. Then, it explains the simulation data of Instance 1 in detail. In Section 4.2, the computational results of the three instances are displayed as well as the timetable for Instance 1. It is proved that the timetable satisfies all the hard constraints.

4.1 Research design

After the mathematical model is developed for the university course timetabling problem, the next step is to examine the performance of this model based on the computation time and the optimal solutions obtained. The sections below introduce the optimization solver and the datasets used to examine the model.

4.1.1 Optimization Solver

FICO Xpress solver Version 8.9 is used for modeling and solving optimization problems. The programming language used here is Mosel language. Its syntax resembles the mathematical equations developed in the last section. It is easy to define sets and parameters. Also, variables can be set as binary by entering the relevant declarations. To model the timetabling problem in Xpress, two files are used: one model file and one Excel data file. The model file contains all the codes of the sets and parameters, the variables, the objective function and the constraints, while the Excel file contains all the data of the parameters. With a large number of data, it is convenient to modify the parameters according to different instances by putting those data in the Excel file. These data are explained below.

4.1.2 Data

The data used for testing the model are generated by the author based on the experience of studying at Wageningen University. Though the computational results are not based on real data, it is sufficient to test the performance of the model. The time length is set as one academic year, which contains six periods and two timeslots per day (morning and afternoon). Therefore, the number of periods is $N_p = 6$ with $s \in \{1,2,3,4,5,6\}$, and timeslots $N_t = 2$ with $t \in \{1,2\}$ (1 denotes "morning" and 2 denotes "afternoon"). For each student group s that takes the same courses, there are 10 courses over the six periods. In periods 1, 2, 5, and 6, two courses are scheduled respectively, one in the morning and the other in the afternoon. In periods 3 and 4, only one course is scheduled respectively. Based on the author's experience, the lectures are scheduled in classrooms in the morning, while in the afternoon, there are group works, which do not have a room or lecturer. The lecturers' workload is set as $M_l = 4$, which means a lecturer can teach at most four courses. Besides, in the objective function, the weights for the two goals are set as $w1 = 0.8$ and $w2 = 0.2$, which means the goal of maximizing lecturers' preferences takes priority.

Table 4.1 shows the main features of the three instances. Instance 1 has the smallest size. Instance 2 has a bit larger size than the first one, which has one more student group, 3 more courses, 14 more lectures, and two more lecturers. The size of Instance 3 is much bigger than the other two.

Table 4.1: Main features of three instances

	Instance 1	Instance 2	Instance 3
Number of courses	18	21	50
Number of lecture	59	73	171
Number of student groups	2	3	8
Number of rooms	6	6	10
Number of lecturers	6	8	20
Number of periods	6	6	6
Number of timeslots	2	2	2

The dataset of Instance 1 is explained here. There are two student groups and 18 courses in total, as is shown in Table 4.2, so there is $C = \{C1, C2, C3, ..., C18\}$. Each student group has 10 courses over the six periods, which are $C_{S1} = \{C1, C2, C3, ..., C10\}$ and $C_{S2} = \{C1, C2, C11, ..., C18\}$. We can see that course $C1$ and $C2$ are the joint courses for both student groups. The third column shows the number of students N_c that attend the course. The fourth column shows the number of lectures Λ_c that each course has per week. On the right side of the table, the room features ($\Phi = \{\Phi1, \Phi2, \Phi3\}$) that the courses require are indicated with a 0-1 matrix. Table 4.3 shows six available rooms ($R = \{R1, R2, ..., R6\}$), with their capacity as well as room features. Table 4.4 shows six available lecturers ($L = \{L1, L2, ..., L6\}$) and their ability to teach courses. Considering the parameters in the objective function, the lecturers' preferences for weekdays and timeslots are indicated in Table 4.5 and Table 4.6 with all the elements ranging from 1 to 5 except the preferences for Friday, which range between 1 and 3 to show a less preference. These numbers are randomly generated by using function "RANDBETWEEN" in Excel. The larger number indicates a higher preference.

Considering constraints, Instance 1 sets some requirements for the courses:

1. Courses not in the same period: For student group 1, course $(C1, C2)$ shall not be scheduled in the same period, so as course $(C3, C4)$. For student group 2, course $(C11, C12)$ shall not be scheduled in the same period, so as course $(C13, C14)$. This refers to constraint 19 (C19).

2. Courses in consecutive periods: For student group 1, course $(C5, C6)$ shall be scheduled in consecutive periods. For student group 2, course $(C15, C16)$ shall be scheduled in consecutive periods. This refers to constraint 20 (C20).

3. Precedenct courses: For student group 1, course $C2$ shall be scheduled before course $C10$, so as courses $(C1, C4)$. For student group 2, course $C11$ shall be scheduled before course $C14$, so as courses $(C15, C18)$. This refers to constraint 21 (C21).

Table 4.2: Courses in instance 1

Courses	Student groups	Number of students	Number of lecture per week	Required room feature		
				Φ1	Φ2	Φ3
C1	S1, S2	40	3	1	1	1
C2	S1, S2	38	2	1	0	0
C3	S1	42	2	1	0	0
C4	S1	45	4	0	1	0
C5	S1	66	3	0	0	1
C6	S1	82	5	1	1	0
C7	S1	77	4	1	0	0
C8	S1	79	4	1	1	0
C9	S1	120	3	1	1	1
C10	S1	101	4	1	1	1
C11	S2	50	3	0	1	1
C12	S2	38	2	0	1	1
C13	S2	36	3	1	0	0
C14	S2	45	3	1	1	1
C15	S2	68	2	1	1	0
C16	S2	46	3	0	1	1
C17	S2	59	4	0	0	1
C18	S2	78	5	0	1	0

0: the course does not require this room feature
1: the course requires this room feature

Table 4.3: Available rooms in instance 1

Room	Capacity	Room feature		
		Φ1	Φ2	Φ3
R1	40	1	1	1
R2	50	1	0	1
R3	60	0	1	1
R4	70	1	1	1
R5	90	1	1	0
R6	120	1	1	1

0: the room does not have this room feature
1: the course has this room feature

Table 4.4: Lecturers' ability to courses in instance 1

	Lecturer					
	L1	L2	L3	L4	L5	L6
C1	1	1	0	1	0	0
C2	1	0	1	0	0	0
C3	0	0	1	0	1	0
C4	0	1	0	0	0	1
C5	1	0	0	1	0	0
C6	0	0	1	0	0	0
C7	0	0	0	1	0	1
C8	0	1	1	0	1	0
C9	1	1	0	1	0	0
C10	0	0	1	0	0	0
C11	0	1	0	1	1	1
C12	0	0	0	0	0	1
C13	0	0	0	1	1	0
C14	0	0	1	0	0	0
C15	1	0	0	1	0	1
C16	0	1	0	0	1	0
C17	0	0	1	0	0	1
C18	0	0	0	0	0	1

0: the lecturer is not able to teach this course.
1: the lecturer is able to teach this course.

Table 4.5: Lecturers' preference for weekdays in instance 1

	Weekday				
Lecturer	Monday	Tuesday	Wednesday	Thursday	Friday
L1	2	5	1	4	3
L2	2	4	4	1	3
L3	5	4	1	3	3
L4	5	4	4	2	3
L5	2	2	2	4	1
L6	1	2	5	1	2

Table 4.6: Lecturers' preference for timeslots in instance 1

	Timeslot	
Lecturer	Morning	Afternoon
L1	3	3
L2	2	1
L3	2	2
L4	4	2
L5	4	3
L6	2	2

4.2 Computational results

This section includes the computational results of running the three datasets in FICO Xpress, as well as the timetables for student groups, lecturers and rooms of Instance 1. As for Instance 2 and 3, due to the large size of the data, the entire results are not presented.

4.2.1 Computational results

Table 4.7 shows the computational results of the three instances after running the three datasets in Xpress. Considering the number of rows and columns, it can be seen that Instance 2 has a bit larger size than Instance 1, while Instance 3 has an exponential increase in size. Considering computation time, Instance 1 and 2 got the optimal solutions after around 60 seconds, while it requires a much longer computation time for Instance 3. Instance 3 still has a gap of 4.44% after around half an hour of computation, and got a currently best solution. This shows that the size of the integer programming model has a large influence on the computation time.

Table 4.7: Computational results of instances in Xpress

	Instance 1	Instance 2	Instance 3
Number of rows (constraints)	18763	25397	240110
Number of columns (variables)	39429	61161	602409
computation time (seconds)	61	44	1644
Objective value (Best solution)	1	0.994	0.975
Best bound	1	0.994	1.018
Solution is optimal	Yes	Yes	No
Gap	0	0	4.44%*

Best solution: the value of the best integer solution found so far.
Best bound: a bound on the value of the optimal integer solution.
Gap: the percentage gap between the best solution and the best bound.
*: the model optimization was stopped by using one of the stopping criteria: "MIPRELSTOP". It means stopping the MIP search when the gap becomes less than the specified percentage value (here it is set as 4.5%).

4.2.2 Timetables for instance 1

According to the optimal solution, a timetable is made for instance 1, which is displayed in Table 4.8. There are two small tables in this timetable. On the left-up side of each small table, "S1" and "S2" denote student group 1 and 2. Students in the same group take the same courses. Therefore, the upper table shows the schedule for students in group 1, while the lower table shows the schedule for students in group 2. In the row of "Morning" and "Afternoon", "x" denotes that a lecture is scheduled on day d. It can be seen that each student group has ten courses over the six periods, with two courses in period 1, 2, 5, and 6, and one course in the morning of period 3 and 4.

Table 4.8: Timetable for student groups in instance 1

S1	Period 1					Period 2					Period 3					Period 4					Period 5					Period 6				
weekday	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Morning	x	x		x		x	x	x	x		x	x	x		x		x		x		x	x	x		x	x		x		x
Course	C5					C8					C7					C3					C4					C9				
Room	R4					R5					R5					R2					R3					R6				
Teacher	L1					L5					L4					L5					L2					L4				
Break																														
Afternoon	x	x		x		x	x	x	x	x												x		x		x	x		x	x
Course	C1					C6															C2					C10				
Room	R1					R5															R1					R6				
Teacher	L1					L3															L1					L3				

S2	Period 1					Period 2					Period 3					Period 4					Period 5					Period 6				
weekday	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Morning	x		x		x	x		x		x	x		x	x		x	x		x		x	x	x	x	x	x	x		x	x
Course	C11					C13					C16					C14					C18					C17				
Room	R3					R1					R3					R4					R5					R3				
Teacher	L4					L4					L5					L3					L6					L3				
Break																														
Afternoon	x	x		x			x		x													x		x		x		x		
Course	C1					C15															C2					C12				
Room	R1					R4															R1					R1				
Teacher	L1					L1															L1					L6				

In addition, to have a clear view on the schedule of lecturers, Table 4.9 shows the courses a lecturer needs to give in a period. Table 4.10 shows the schedule for each room in each period.

Table 4.9: Lecturers' schedule

	Priod, Timeslot										
Lecturer	P1 MO	P1 AF	P2 MO	P2 AF	P3 MO	P4 MO	P5 MO	P5 AF	P6 MO	P6 AF	Sum of courses
L1	C5	C1		C15				C2			4
L2							C4				1
L3				C6		C14			C17	C10	4
L4	C11		C13		C7				C9		4
L5			C8		C16	C3					3
L6							C18			C12	2
Sum of courses	2	1	2	2	2	2	2	1	2	2	18

Table 4.10: Schedule for rooms

	Priod, Timeslot (Number of students)										
Room (Capacity)	P1 MO	P1 AF	P2 MO	P2 AF	P3 MO	P4 MO	P5 MO	P5 AF	P6 MO	P6 AF	Sum of courses
R1 (40)		C1 (40)	C13 (36)					C2 (38)		C12 (38)	4
R2 (50)						C3 (42)					1
R3 (60)	C11 (50)				C16 (46)		C4 (45)		C17 (59)		4
R4 (70)	C5 (66)			C15 (68)		C14 (45)					3
R5 (90)			C8 (79)	C6 (82)	C7 (77)		C18 (78)				4
R6 (120)									C9 (120)	C10 (101)	2
Sum of courses	2	1	2	2	2	2	2	1	2	2	18

The feasibility of the generated timetables is checked with the constraints in Table 3.3 in Section 3.2 and detailed data in Section 4.1.2. Summarized characteristics of the timetables are:

1. Completeness: This timetable includes 18 courses for two student groups, with ten courses for each group (C13). In the afternoon of period 3 and 4, no lectures are scheduled (C15). Each course is assigned a lecturer, a room and a timeslot. 59 lectures in total are scheduled and all the courses have the required number of lectures per week (C1).
2. No conflicts: Each room holds at most one course at a specific timeslot, and a lecture is assigned with exactly one room (C2 & C3). A lecturer teaches at most one course at a specific time, and all the lectures of a course are taught by the same lecturer (C7, C8 & C9). Within a student group, there are no time conflicts among the courses (C12).
3. Room capacity: In Table 4.10, it can be seen that the room assigned to each course has sufficient seats (C5).
4. Lectures of each course are spread out over the week (C16, C17 & C18).
5. Room stability: All the lectures of a course are held in the same room (C4).
6. Time consistency: All the lectures of a course are scheduled at the same timeslot on different days and in the same period (C14).
7. Lecturers' workload: The rightmost column of Table 4.9 shows that the workload of each lecturer does not exceed the maximum load 4 (C11).
8. Lecturers' ability: Table 4.11 below shows each course is matched a lecturer who is able to teach it (C10). Colored cells denote the matches.
9. The timetable satisfies the detailed constriants in Section 4.1.2 (C19, C20, C21).
10. Some courses require specific room features or equipment (C6). This can be checked in Table 4.12 below.

In conclusion, the extended model is able to find the optimal solution that satisfies all the prescribed requirements.

Table 4.11: Visualization of lecturers' ability to courses in instance 1

	Lecturer					
	L1	L2	L3	L4	L5	L6
C1	1	1	0	1	0	0
C2	1	0	1	0	0	0
C3	0	0	1	0	1	0
C4	0	1	0	0	0	1
C5	1	0	0	1	0	0
C6	0	0	1	0	0	0
C7	0	0	0	1	0	1
C8	0	1	1	0	1	0
C9	1	1	0	1	0	0
C10	0	0	1	0	0	0
C11	0	1	0	1	1	1
C12	0	0	0	0	0	1
C13	0	0	0	1	1	0
C14	0	0	1	0	0	0
C15	1	0	0	1	0	1
C16	0	1	0	0	1	0
C17	0	0	1	0	0	1
C18	0	0	0	0	0	1

Table 4.12: Visualization of room features for courses in instance 1

	Feature				Required feature		
Room	Φ1	Φ2	Φ3	Course	Φ1	Φ2	Φ3
R1	1	1	1	C1	1	1	1
				C2	1	0	0
				C12	0	1	1
				C13	1	0	0
R2	1	0	1	C3	1	0	0
R3	0	1	1	C4	0	1	0
				C11	0	1	1
				C16	0	1	1
				C17	0	0	1
R4	1	1	1	C5	0	0	1
				C14	1	1	1
				C15	1	1	0
R5	1	1	0	C6	1	1	0
				C7	1	0	0
				C8	1	1	0
				C18	0	1	0
R6	1	1	1	C9	1	1	1
				C10	1	1	1

Chapter 5

Discussion

5.1 Normalization of the multi-objective function

In this model, the weighted sum method and the normalization method are used for the multi-objective function. This is different from the reviewed articles. Algethami & Laesanklang (2021) and Arratia-Martinez, Maya-Padron & Avila-Torres (2021) developed the multi-objective function by simply summing all the goals. Colajanni (2019) added a penalty weight to each goal, but without normalizing those goals that have different scales and units. In Section 3.3.3, the max-min normalization method is used on the two goals f_1 and f_2. The maximum and minimum values are calculated for each goal. The maximum and minimum values of f_1 and f_2 in Instance 1 are shown in Table 5.1 below. A drawback of using this method in this model is that it requires a long computation time to calculate some of the extreme values with a relatively small gap, especially with the large instance. But since the gap is small enough, it may not have a large influence on the normalization results.

Table 5.1: Maximum and minimum values of f_1 and f_2 in Instance 1

	value
f_1^{max}	376
f_1^{min}	249
f_2^{max}	-535
f_2^{min}	-2655

In three instances, the weights are all set as $w_1 = 0.8$ and $w_2 = 0.2$, which shows a priority for lecturers' preferences. A finding is that in Instance 1, when changing w_1

and w_2 while keeping $w_1 + w_2 = 1$, the optimal objective value of f_1 and f_2 remain the same, except for the last example, where the value of f_1 is slightly lower. This is shown in Table 5.2. Comparing this table with Table 5.1, it can be seen that the optimal objective value of f_1 and f_2 are the same as the maximum value of both goals. This indicates that in Instance 1, the two goals are not contradictory. This is possible, though it rarely happens. However, in Instance 2, the optimal objective values of f_1 and f_2 vary based on the different weights. This can be observed in Table 5.3. As for Instance 3, due to the long computation time, the values of f_1 and f_2 based on different weights are not presented here.

Table 5.2: Values of f_1 and f_2 in Instance 1

w_1	w_2	f_1	f_2
0.999	0.001	376	-535
0.99	0.01	376	-535
0.8	0.2	376	-535
0.5	0.5	376	-535
0.3	0.7	376	-535
0.01	0.99	376	-535
0.001	0.999	372	-535

Table 5.3: Values of f_1 and f_2 in Instance 2

w_1	w_2	f_1	f_2
0.8	0.2	615	-1016
0.7	0.3	614	-956
0.5	0.5	614	-956
0.3	0.7	614	-956
0.2	0.8	599	-886

5.2 Performance of the extended model

The performance of the university course timetabling model is measured based on the solution quality and computation time. On one hand, the timetables for the three instances meet all the hard constraints. Instance 1 and 2 got the optimal solution which means satisfying the soft constraints as well as possible, while Instance 3 got a good quality solution after a certain time. As the model is a tool to help people make good decisions, for practical use of such a model, it should consider both the solution quality and costs (such as computation time). Sometimes, it is computationally infeasible to get an optimal solution due to the complexity of the problem. In this thesis, considering the goal of maximizing lecturers' preferences, it is not necessary to get an optimal solution. It is better to get a good feasible solution within an acceptable time. Therefore, for Instance 3, it is acceptable to still have a gap of 4.44%. Though it is not optimal, it still provides a workable timetable that meets most of the lecturers' preferences.

When running the extended model with large instances in FICO Xpress, five stopping criteria can terminate the computation instead of waiting for the optimal solution. These criteria are:

1. TIMELIMIT: The maximum time in seconds that the Optimizer will run before it terminates.
2. SOLTIMELIMIT: A soft limit (in seconds) on runtime for an integer programming solve. The solver stops whenever a feasible MIP solution is found and the runtime exceeds the specified value.
3. MIPRELSTOP: Stop the integer programming search when the gap between the current best solution and the current best bound becomes less than the specified percentage.
4. MIPABSSTOP: Stop the integer programming search when the gap between the current best solution and the current best bound becomes less than the specified absolution value.
5. MAXNODES: The maximum number of nodes that will be explored by branch and bound.

Among these five stopping criteria, "SOLTIMELIMIT" and "MIPRELSTOP" are recommended. Compared with "TIMELIMIT", "SOLTIMELIMIT" guarantees a feasible solution before the solver stops. It is more straightforward to set a percentage difference ("MIPRELSTOP") than set a specific number ("MIPABSSTOP"), as the percentage difference shows more apparently that relatively how far the solution is from the best bound with "MIPABSSTOP". Finally, it depends on the decision maker, that how long the computation or how good the solution quality is desired.

On the other hand, considering computation time, small-scale instances, like Instance 1 and 2, can be solved to optimality within a short time, while with the problem size increasing, a much longer computation time is required. This result aligns with those in literature. One explanation could be that a set of constraints concerning room stability increases the computational complexity, thus imposing a significant negative impact on computation time (Lach & Lübbecke, 2012). Phillips et al. (2015) conducted a research on the classroom assignment problem in university course timetabling and discussed the impact of room stability constraints on computational complexity. According to their opinion, the most simple room assignment problem is to assign a set of available rooms to a set of lectures (courses), just like the basic model formulated in Section 3.1. With this formulation, each time period can be treated as an independent assignment problem and the constraint matrix is totally unimodular, so it can be solved efficiently. However, in practice, when a lecture covers more than one timeslot, usually the lecture shall be scheduled in the same room within all the consecutive timeslots. Under this circumstance, the formulation changes the structure of the constraint matrix, leading to the occurrence of fractions in the matrix. Therefore, the Linear Programming relaxation cannot guarantee to be integer, which leads to a long computation time. This makes the problem NP-hard, which means, with the problem size and complexity increasing exponentially, it cannot be solved to optimality within polynomial time (Babaei, Karimpour & Hadidi, 2015).

There is a similar situation in the extended model. Parameter Λ_c denotes the number of lectures per week for course c, and there shall be at most one lecture per day for each course. Constraint 14 rules that all the lectures of a course in a week shall be scheduled in the same room. Now, each weekday can be assumed as a single timeslot and the whole week assumed as a day. This resembles that there are five timeslots per day. Therefore, the extended model requires a long time to compute the optimal solution. An experiment is conducted on Instance 1 and 2. When constraint 14 is removed, both instances get optimal with 10.8 seconds and 12.6 seconds, while before they require 61 seconds and 44 seconds to get optimal.

In addition, according to the experimental experience of MirHassani (2006) and Daskalaki, Birbas & Housos (2004), the computation time can be varied largely by changing the coefficient values. This finding is also proved with Instance 1 and 2. By changing lecturers' preferences $P1_{k,d}$ and $P2_{k,t}$, the computation time varies a lot, from 1 second to 300 seconds. Therefore, the coefficient value is another factor that can impact the computation time.

Another advantage of the extended model is that it is applicable to different situations by making adjustments to the model. It is convenient to modify the parameters through Excel files in an easy way. For example, it is not only able to deal with one academic year, but also with any number of periods by changing p in Excel. Also, the information of rooms and courses can be modified in Excel, as well as the preferences of lecturers. The model can generate different solutions based on those modifications.

Chapter 6

Conclusions, limitations and further research

6.1 Conclusions

In this report, an integer programming model is developed for the university course timetabling problems that are inspired by the real situation of master programs at Wageningen University. The model is tested successfully in FICO Xpress with three instances. The results show that the generated timetables satisfy all the hard constraints. In addition, by considering lecturer preferences, the model results in timetables that better meet the requirements of lecturers. Also, this model ensures better utilization of resources. It helps optimize the allocation of classrooms by avoiding assigning large rooms to courses with a small number of students as much as possible.

It is found that the problem size of university course timetabling has a large influence on computation time. With small instances, the model can provide an optimal solution, while with large instances, it can provide a good feasible solution within an acceptable time. Another interesting finding is that the coefficient values of the objective functions could have a large impact on the computation time. Future research could focus on exploring the relationships between computation time and coefficient values. Besides, it is a flexible and general model that can be applied to different situations by making adjustments to the model or changing the parameters easily in Excel.

6.2 Limitations and future research

There are also some limitations concerning this model. First, while it is important to have a comprehensive and accurate model, it is also important that the model can provide good feasible or even optimal solutions efficiently. In terms of the current model, the size of the problem imposes certain difficulties in attaining an optimal solution. Therefore, future research could focus on simplifying the model by reducing its complexity, thus reducing computation time while at the same time, not compromising solution quality.

Second, this model still leaves a gap between theory and real-world situations. On one hand, the university timetabling problem is based on the literature review as well as the author's experience of studying at Wageningen University. To make it more practical, it is recommended to develop a model based on the real situation of Wageningen University. On the other hand, currently, the model contains some assumptions to simplify the problem. For example, it defaults that there are only lectures for each course, but in practical situations, there could also be tutorials and lab work. There could be requirements on the sequence of theory lectures and tutorials. For instance, the lectures shall be scheduled before tutorials. Another example is, at Wageningen University, there are situations when more than one lecturer teaches a course, or even a lecture. However, the current model fails to embrace these characteristics. Therefore, future research could focus on embracing more rules and requirements to make it more sophisticated and more tailored to the real-world situation. In addition, real-world problems are often subject to uncertainty, and the timetabling problem is no exception. Future research could focus on incorporating uncertainty into the model, for example by considering the possibility of unexpected events that may disrupt the timetable.

Last, in this thesis, the model is tested with simulation datasets created by the author. However, it is recommended to collect real data to improve the accuracy and effectiveness of the model. In addition, to modify this model, it requires advanced knowledge of FICO Xpress and an advanced understanding of the model itself. However, it is important to ensure that the model can be easily modified and used by others. Therefore, future research could focus on developing a user-friendly interface to improve the utilization. This would make it easier to make adjustments to the data and view the generated timetable directly.

Bibliography

Abidi, H. Leeuw, S. & Klumpp, M. (2014) *Humanitarian supply chain performance management: a systematic literature review*. Supply Chain Management: An International Journal, 19, 592 - 608 http://dx.doi.org/10.1108/SCM-09-2013-0349

Algethami, H. & Laesanklang, W. (2021). *A Mathematical Model for Course Timetabling Problem With Faculty-Course Assignment Constraints*. in IEEE Access, 9, 111666-111682 https://doi.org/10.1109/ACCESS.2021.3103495. [2]

Anagnostopoulos, k.p. & Mamanis, G. (2010) *A portfolio optimization model with three objectives and discrete variables* Computers & Operations Research, 37(7), 1285-1297 https://doi.org/10.1016/j.cor.2009.09.009

Arratia-Martinez, N.M., Maya-Padron, C. & Avila-Torres, P.A. (2021) *University Course Timetabling Problem with Professor Assignment*. Mathematical Problems in Engineering, vol. 2021, Article ID 6617177, 9 pages. https://doi.org/10.1155/2021/6617177. [8]

Aziz, N. & Aizam, N. (2018). *A brief review on the features of university course timetabling problem*. AIP Conference Proceedings 2016, 020001 (2018). https://doi.org/10.1063/1.5055403

Babaei, H., Karimpour, J. & Hadidi, A. (2015). *A survey of approaches for university course timetabling problem*. Computers and Industrial Engineering, 86, 43-59. https://doi.org/10.1016/j.cie.2014.11.010

Bakir, M. A. & Aksop, C. (2008). *A 0-1 integer programming approach to a university timetabling problem*. Hacettepe Journal of Mathematics and Statistics, 37(1), 41-55. [3]

Ceschia, S., Di Gaspero, L. & Schaerf, A. (2022). *Educational timetabling: Problems, benchmarks, and state-of-the-art results*. European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2022.07.011

Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N. R., Kendall, G. (2021) *A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities*. in IEEE Access, 9, 106515-106529 https://doi: 10.1109/ACCESS.2021.3100613.

Colajanni, G. (2019). *An Integer Programming Formulation for University Course Timetabling*. Advances in Optimization and Decision Science for Society, Services and Enterprises. 219-231. [4]

Daskalaki, S., Birbas, T. & Housos, E. (2004). *An integer programming formulation for a case study in university timetabling*. European Journal of Operational Research, 153(1), 117-135. https://doi.org/10.1016/S0377-2217(03)00103-6 [5]

Domenech, B. & Lusa, A. (2016) *A MILP model for the teacher assignment problem considering teachers' preferences*. European Journal of Operational Research, 249(3), 1153-1160. https://doi.org/10.1016/j.ejor.2015.08.057. [6]

Fonseca, G. H. G., Santos, H. G., Carrano, E, G., Stidsen, T. J. R. (2017). *Integer programming techniques for educational timetabling*. European Journal of Operational Research, 262(1), 28-39. https://doi.org/10.1016/j.ejor.2017.03.020

Havås, J., Olsson, A., Persson, J., Schierscher, M.S. (2013). *Modeling and optimization of university timetabling-a case study in integer programming*. [7]

Lach, G. & Lübbecke, M., (2012). *Curriculum based course timetabling: new solutions to Udine benchmark instances*. Ann Oper Res, 194, 255–272 https://doi.org/10.1007/s10479-010-0700-7

Lawrie, N. L. (1969) *An integer linear programming model of a school timetabling problem*. The Computer Journal, 12, 307–316

Mikkelsen, R. (2021). *Application of Mixed Integer Programming Methods for Practical Educational Timetabling*. [Doctoral thesis, Technical University of Denmark]. DTU Orbit. https://backend.orbit.dtu.dk/ws/portalfiles/portal /271654196/Rasmus-Mikkelsen_PhD_Thesis.pdf

Mirhassani, S. & Habibi, F. (2013). *Solution approaches to the course timetabling problem*. Artificial Intelligence Review, 39(2), 133-149. https://doi.org/10.1007/s10462-011-9262-6

Mirhassani, S. (2006) *A computational approach to enhancing course timetabling with integer programming*. Applied Mathematics and Computation, 175(1), 814-822 https://doi.org/10.1016/j.amc.2005.07.039. [9]

Phillips, A., Waterer, H., Ehrgott, M., Ryan, D. (2015) *Integer programming methods for large-scale practical classroom assignment problems*. Computers & Operations Research, 42-53. https://doi.org/10.1016/j.cor.2014.07.012

Schaerf, A. (1999). *A Survey of Automated Timetabling*. Artificial Intelligence Review, 13, 87-127. https://doi.org/10.1023/A:1006576209967

Schaerf, A. & Gaspero, L. (2007). *Measurability and Reproducibility in University Timetabling Research: Discussion and Proposals*. In: Burke, E.K., Rudová, H. (eds) Practice and Theory of Automated Timetabling VI. PATAT 2006. Lecture Notes in Computer Science, vol 3867 https://doi.org/10.1007/978-3-540-77345-0_3

Schimmelpfeng, K. & Helber, S. (2007). *Application of a real-world university-course timetabling model solved by integer programming.* OR Spectrum, 29, 783–803 https://doi.org/10.1007/s00291-006-0074-z

Suárez-Rodríguez, J. et al. (2021). *An Optimization Model for University Course Timetabling-A Colombian Case Study*. Communications in Computer and Information Science, vol 1407. [1]

Zaulir, Z., Aziz, N. & Aizam, N. (2022) *A General Mathematical Model for University Courses Timetabling: Implementation to a Public University in Malaysia*. MJFAS, 18(1), 82-94. https://doi.org/10.11113/mjfas.v18n1.2408. [10]

Appendix A

Systemiatic literature research

In this thesis, a systematic literature review was conducted. This chapter preseants the search approaches and search results. Search approaches include two steps, searching and screening. The search strategies shows which search words and Boolean operators were used and combined to obtain the required literature. The search criteria set the requirements to screen the publications. Finally, the results of the search are summarized.

A.1 Search approaches

A.1.1 Searching

Based on the research questions illustrated in Chapter 1, key terms are identified as well as their synonyms to identify and evaluate the literature. These terms are summarized in Table A.1 below. The identified search terms are then used to create search strings with Boolean operators (AND, OR, AND NOT). Table A.2 presents the used databases, the search scope, the search date, and the number of publications of each search. There are 2086 papers in total.

A.1.2 Screening

After the general literature search, a set of search criteria was developed to clarify the inclusion and exclusion of papers. The criteria are listed below:

1. Papers should be written in English.

Table A.1: Key terms and their syninyms

Key terms	Synonyms
timetabling	timetable
university course timetabling problem	educational timetabling; university timetabling; course timetabling
integer programming model	0-1 integer programming; IP; integer linear programming; mathematical model; integer model
mixed integer programming	MIP
room assignment	classroom assignment
review	survey; overview; summary

Table A.2: Database search

Database	Scope	Date of search	Number of publications
Scopus	title, abstract and keywords	17/11/2022	25
Scopus	title, abstract and keywords	17/11/2022	31
Google Scholar	all fields	23/11/2022	2030
total			2086

2. Papers should be academic literature, but some scientific reports can also be reviewed.

3. Papers published after 2000 will be included.

4. Papers related to the overview of university course timetabling problems will be included.

5. Papers related to sovling university course timetabling problems through integer programming will be included. Considering that mixed integer programming is similar to integer programming in the formulation of equations, papers related to using mixed integer programming to solve university course timetabling problems may also be included.

6. Papers using non-exact methods will be excluded. Some articles use heuristics or hybrid methods to look for feasible solutions for the integer programming models.

These articles will be excluded, as has mentioned in Chapter 1, this thesis only focus on exact methods.

7. Parers using advanced methods to the integer programming models will be excluded, such as decomposition, reformulation.

Based on the screening criteria, 14 papers are selected after a full paper analysis. Figure A.1 shows the process of screening the papers. The useful publications are listed in Table A.3.

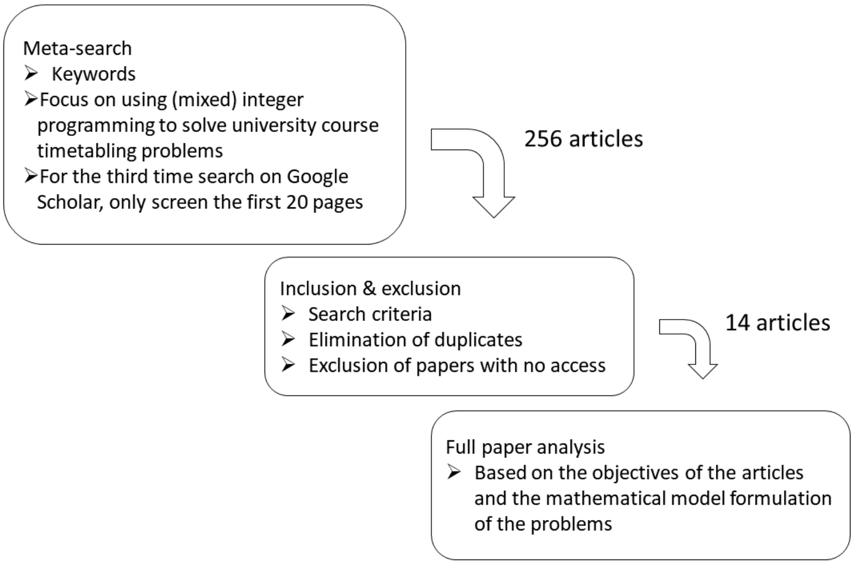

Figure A.1: Literature screening

Table A.3: Search strategies

1	search strings	database
	(TITLE-ABS-KEY ("university course timetabl*" OR "university timetabl*" OR "course timetabl*" OR "educational timetabl*") AND TITLE-ABS-KEY ("survey"))	Scopus
search results: 2 useful papers		
Babaei, Karimpour & Hadidi (2015); MirHassani & Habibi (2013)		
2	**search strings**	**database**
	(TITLE-ABS-KEY ("university course timetabl*" OR "university timetabl*" OR "course timetabl*" OR "educational timetabl*") AND TITLE-ABS-KEY ("review" OR "summary" OR "overview"))	Scopus
search results: 1 useful papers		
Chen et al. (2021)		
3	**search strings**	**database**
	("university timetabling" OR "university course timetabling") AND ("integer programming" OR "mixed integer programming" OR "IP" OR "MIP")	Google Scholar
search results: 11 useful papers		
Daskalaki, Birbas & Housos (2004); Bakir & Aksop (2008); Aziz & Aizam (2018); Algethami & Laesanklang (2021); Domenech & Lusa (2016); Zaulir, Aziz & Aizam (2022); Arratia-Martinez, Maya-Padron & Avila-Torres (2021); Colajanni (2019); Havås et al. (2013); MirHassani (2006); Suárez-Rodríguez et al. (2021)		

49

A.2 Search results

In this section you will find the articles that have been used for the systematic literature review. The articles mentioned below have been found in Scopus and Google Scholar. In Chapter 2, ten articles on developing university course timetabling models through (mixed) integer programming were reviewed and the requirements of the timetabling problems in the articles were listed and identified as hard or soft constraints. In Table A.4, the useful data are extracted from the papers. The first column the authors and year are mentioned, furthermore the aim of the study and the useful findings for answering the first research question.

Table A.4: Search results

Paper	Aim	Main findings
Babaei, Karimpour & Hadidi (2015)	A detailed review of solution approaches to solving university course timetablings	This paper defined the classfication of different solution approaches and stated that intger programming falls under operational research methods.
MirHassani & Habibi (2013)	A review focusing on the researches of university course timetabling from 2002 and on the modelling based methods	This paper defined hard and soft constraints and listed a set of constraints used in other researches.
Chen et al. (2021)	A survey on the most recent approaches in solving university timetabling problems and case studies of the real-world problems	This paper defined the classfication of different solution approaches and stated that intger programming falls under operational research methods.
Daskalaki, Birbas & Housos (2004)	Using a novel 0-1 integer programming model to solve a university course timetabling problem.	The model involves elements of lecturers, students, days, timeslots, courses and rooms. The constraints contain five aspects; which are 1. Avoiding conflicts among time, rooms, lecturers and students. 2. Completemess: all the lecture in a timetable shall be assigned. 3. Consecutiveness: if an lecture covers more than one timeslot on a day, these timeslots shall be consecutive. 4. Repetitiveness: if an lecture is designed for small groups, it shall be repeated several times to accommodate all the students registered for it. 5. Pre-assignment: the pre-assigned lecture shall not be violated. It uses a set of basic variables and a set of auxiliary variables. The latter ones are used to enforce consecutiveness. The objective function is to minimize the cost of assigning courses.

Continue on the next page

Table A.4: Sesrch results (cont.)

Paper	Aim	Main findings
Bakir & Aksop (2008)	Formulating a 0-1 integer programming model to solve the existed timetabling problem in Daskalaki, Birbas & Housos (2004).	The model involves elements of lecturers, students, days, timeslots, courses and rooms. The constraints constain 1. Avoiding conflicts among students, courses and rooms. 2. Meeting credit requirements. 3. Consecutiveness: if an lecture covers more than one timeslot on a day, these timeslots shall be consecutive. 4. Room consistency: if an lecture covers more than one timeslot, it shall be given in the same room over the timeslots. 5. Precedence: theory lectures shall be given before practice. 6. Pre-assignment: the timeslots for the pre-assigned lecture shall not be occupied. 7. Time restrictions: timeslots in the late of a day shall be avoided. 8. Considerations for failed students. The objective is to minimize the dissatisfaction of students and lecturers.
Aziz & Aizam (2018)	A brief review on the characteristics of university course timetabling problems	This paper defines the elements of course timetabling problems, which are time, room, lecturer, students and course.
Algethami & Laesanklang (2021)	Developing an automated timetable through mixed integer programming with a focus on faculty-related constraints.	The model involves elements of courses, days, timeslots, rooms, lecturers and student groups. The constraints include 1. Avoiding conflicts among lecture, rooms, lecturers, students and time. 2. Distribution of courses: courses with more credits shall be assigned in the morning. 3. Meet teaching credit requirements 4. Time restrictions: An hour break shal be scheduled at noon per day. 5. Workloads of faculty members. 6. Room stability. 7. Room capacity. 8. Capability of lecturers: a lecturer can only teach courses s/he is able to teach. 9. Room features: Courses shall be assgined to the rooms which have the required equipment or features. 10. Availability of lecturers at certain timeslots. The objective is to maximize lecturers' preferences and minimize the total days a student have to visit the campus.

Continue on the next page

52

53

Table A.4: Sesrch results (cont.)

Paper	Aim	Main findings
Domenech & Lusa (2016)	Developing a mixed integer programming model to solve lecturer-course assignment problems in a Spanish university.	The model involves elements of lecturers, timeslots, semesters and courses. The constraints include: 1. Avoiding conflicts among lecturers and timeslots. 2. A course can only be taught by one lecturer. 3. Availability of lecturers: a lecturer can only teach courses s/he is able to teach. The objective is to maximize lecturers' preferences and balance lecturers' workloads.
Zaulir, Aziz & Aizam (2022)	Developing a general integer programming model suitable for most course timetabling problems for universities in Malasia	The constraints include completeness, room capacity, availability of lecturer, time and room, avoiding conflicts, workloads of lecturers and students, distribution of lectures (consecutive or interval), time restrictions, precedence. The objective is to maximize lecturers preferences.
Arratia-Martinez, Maya-Padron & Avila-Torres (2021)	Presenting a new mathematical model for a specific university department in Mexico through integer programming focusing on lecturer-course-time assignment.	The model involves elements of lecturers, courses, timeslots and course sections. The constraints include: 1. Completeness: all the courses and their sections are scheduled. 2. Avoiding conflicts among courses, lecturers and time. 3. Non-overlapping: each course section is assigned once. 4. Workloads of lecturers. 5. Capability of lecturers: lecturers can only teach courses that they are able to teach. 6. Time restictions: some courses can only be scheduled in the morning and the others in the afternoon. 7. Availability of lecturers: full-time lecturers shall have the same free time for administrative activities, and some timeslots are preserved for other lecture. The objective is to minimize the number of courses that are not assigned a lecturer. Another objective is to balance the number of courses scheduled in each timeslot.

Continue on the next page

Table A.4: Sesrch results (cont.)

Paper	Aim	Main findings
Colajanni (2019)	Proposing a new integer programming model to solve the timetabling problem for a depatment in University of Catania.	The model involves elements of courses, rooms, timeslots, days, lecturers, students. The constraints inlcude: 1. Completeness: all the lessons of a course shall be scheduled. 2. Avoiding conflicts among lessons, time, lecturers and rooms. 3. Availability of lecturers: a course shall not be scheduled to the timeslot when the lecturer is not available. 4. Room features: assigning courses to speific rooms with required teaching tools. The objective concerns several aspects. First, the timetable shall be compact as well as psooible. Second, lessons of the same course shall not be scheduled on consecutive days. Third, the number of students of a course shall be close to the capacity of the assigned room to minimize the waste of space. Forth, room stability shal be fulfilled as much as possible.
Havås et al. (2013)	Developing a course timetabling model for a department of a German university through integer programming	The model involves elements of courses, days, timeslots and rooms. The constraints include: 1. Avoiding conflicts among lecture, rooms and time. 2. Room capacity. 3. Room stability. 4. Distribution of lecture: no more than one lecture, exercise and computer lab are assigned per day per course. 5. Compactness of the timetables for different student groups. 6. Room features: there are different rooms suitable for lectures, exercises and computer labs. 7. Precedence: an exercise shall be scheduled after a lecture for the same course. 8. Meeting credits requirements: each course shall have the required number of lecture. The objective is to minimize the dissatifaction with timeslots.

Continue on the next page

54

Table A.4: Sesrch results (cont.)

Paper	Aim	Main findings
MirHassani (2006)	Developing a novel 0-1 integer programming model to satisfy the constraints found in most academic institutions.	The model involves elements of lecturers,courses, days and timeslots. The hard constraints include: 1. Avoiding conflicts among lecturers, students and courses 2. Workloads of lecturers and students. 3. Pre-assigned subjects. The objective is to minimize the non-preferred timeslots, and meet the soft constraint as much as possible. The soft constraint is to have one day off between two sessions of a course in a week.
Suárez-Rodríguez et al. (2021)	Proposing a mixed integer model for the course timetabling problem in a Colombian University.	The model involves elements of timeslots, day, courses, lecturers, semesters, rooms. The constraints includes the folowing aspects: 1. Completeness: all the credit hours are scheduled. 2. Distribution of lectures: the sessions of a course in a week cannot be assigned on the same day and there shall be at least one day in between different sessions. 3. Non-overlapping: courses belonging to the same semester do not overlap; lecturers' availability at some timeslots. 4. Availability of time: Friday afternoon is pre-occupied by other lecture. 5. Meeting credit requirements. 6. Avoiding conflictsamong lecturer, course, time and room. 7. Capability of lecturers: lecturers can only teach courses that they are able to teach. 8. Consideration of workloads for full-time and part-time lecturers. 9. Room capacity. The objective is to minimize the amount of time that lecturers miss to fulfill their workload.

55