POLITECNICO DI TORINO

Corso di Laurea in Communications and Computer Networks Engineering

Tesi di Laurea Magistrale

A Scalable Graph-based Mixed-Integer Linear Programming Approach for the Examination Timetabling Problem

Relatore
Prof. Roberto TADEI

Correlatori:
Prof. Marius PESAVENTO
M.Sc. Ganapati HEGDE

Tianyi LIU
matricola: 266876

ANNO ACCADEMICO 2017 – 2018

Summary

In this thesis the Examination Timetabling Problem (ETP) in the Darmstadt University of Technology (TU Darmstadt) is presented and a Mixed-Integer Linear Programming (MILP) model is proposed for it. Our model concentrates on the conflicts of students. An exam-based conflict graph in which edges represent incompatibilities between exams is used. An exact MILP approach is directly using a MIP solver to solve the model, which is usually not able to solve real instances due to the complexity. In order to achieve high-quality solutions within a short computational time, we propose a scalable approach based on decomposing the entire problem into subproblems, which can be easily handled using the exact MILP approach. This approach concentrates on dealing with the conflicts. The decomposition of the problem then corresponds to the decomposition of the conflict graph. For the test instances in this thesis, the scalable approach considerably improves the solutions even in shorter time, compared with the exact MILP approach.

Contents

List of Tables

List of Figures

Glossary of Symbols

Acronyms

ETP	Examination Timetabling Problem
MIP	Mixed-Integer Programming
MILP	Mixed-Integer Linear Programming
MCP	Maximum Clique Problem
MWCP	Maximum Weight Clique Problem
MWQCP	Maximum Weight Quasi-Clique Problem
MWQCP1	Maximum Weight Quasi-Clique Problem with quasi-clique definition 1
MWQCP2	Maximum Weight Quasi-Clique Problem with quasi-clique definition 2

Chapter 1

Introduction

The Examination Timetabling Problem (ETP) is a type of educational timetabling problem which has been well researched. It deals with the assignment of exams to timeslots and rooms, subject to a set of constraints. One of the most common constraints is to avoid *conflicts*, i.e., no student is required to write two exams at the same time. This difficult combinatorial problem must be faced by educational institutions at the end of each semester. ETP problems are proven to be NP-complete. Solving a real-world ETP problem manually often requires a large amount of time, expensive resources, as well as administrative experience.

Educational institutions may consider different constraints, which result in many variants of ETP problems. In this thesis, we discuss the particular ETP problem in the Darmstadt University of Technology (TU Darmstadt). The post-enrollment model is employed. That is, the problem contains the information of students' enrollments. Due to the complexity of this information, the traditional manual timetabling usually considers only the curriculum of degree programs, whose deficiencies were discussed in [Bergmann et al., 2014].

In this thesis, we first propose a Mixed-Integer Linear Programming (MILP) model for our ETP problem. An exact approach is directly using a MIP solver to solve the model. The MIP solver used in this thesis is CPLEX of version 12.8.0 and the software is the IBM ILOG CPLEX Optimization Studio (CPLEX Studio), which are introduced in Chapter 5. However, for real instances of this problem, MIP solvers are not able to find an optimal or high-quality solution in reasonable time. Besides, if only a small part of the entire problem is considered, then the exact MILP approach is still considerable. Therefore, we propose a scalable approach that decomposes the entire ETP problem into a hierarchy of subproblems. Each subproblem considers only a subset of exams and is handled using exact MILP approach. An exam-based conflict graph in which edges represent incompatibilities between exams is introduced. The decomposition of the problem then corresponds to the decomposition of the conflict graph. This approach is referred to as *hierarchical construction approach* in this thesis, as it constructs the timetable, layer by layer, by solving a sequence of subproblems. Moreover, the hierarchical construction approach concentrates on dealing with conflicts.

This thesis is structured as follows. In Chapter 2 we present a brief review and discussion of the research on examination timetabling. The main aim of this chapter is to illustrate the diversity of solution methods used in research of timetabling, which may potentially inspire future research on our problem. Chapter 3 formally presents the ETP problem in TU Darmstadt and the mathematical formulation of the MILP model. Chapter 4 describes the hierarchical construction approach and the decomposition techniques employed on the conflict graph. Results, in Chapter 5, show that the hierarchical construction approach considerably improves the solutions to the entire problem even in shorter time, compared with the exact MILP approach. Our conclusion, together with some issues and potential future works, is to be found in Chapter 6.

Chapter 2

Background Works

In this chapter, we present a brief review and discussion of the research on examination timetabling. We first discuss the definitions of the ETP problem from the literature and the complexity of problem. Then, we provide an overview of the solution methods that appeared in the literature. The main aim of this chapter is to illustrate the diversity of solution methods used in research of timetabling, which may potentially inspire future research on our problem.

[Qu et al., 2009] is a very comprehensive survey on examination timetabling and it provides an exhaustive list of surveys on educational timetabling and also summarizes the techniques widely used in the research of timetabling prior to it. We recommend the reader to refer to this paper for additional details on the research of exam timetabling.

The similarity between different types of timetabling problem has been declared in many research papers, since all can be considered as assigning events (exams) to timeslots and resources (e.g., rooms). Thus, this chapter focuses on exam timetabling, while some relevant research and techniques in other types of timetabling are also included.

2.1 Overview on examination timetabling

In [Qu et al., 2009] the Examination Timetabling Problem (ETP) is defined as assigning a set of exams into a finite number of timeslots (periods) and rooms, subject to a set of constraints.

Since different educational institutions may have different rules and expectations, a large variety of constraints exist in many variants of ETP problems and some of them may even contradict each other. This causes a heavy challenge in the research on exam timetabling. The constraints are usually categorized into two types: *hard constraints* and *soft constraints*. The hard constraints cannot be violated under any circumstances, while the soft constraints are desirable and are not absolutely critical. A solution that satisfies all of the hard constraints is usually said to be *feasible*. Then, the violation of soft constraints is used to measure the quality of a feasible solution, so the objective is to minimize this violation. In practice, it is usually impossible to find a feasible solution that satisfies all the soft constraints. Usually many different soft constraints are considered even in one variant of ETP problems, so ETP problem is typically a multi-objective problem.

Some concepts in regard to the conflict in timetable are given first. The term a pair of *conflicting* exams or exams with *conflict potential* refers to a pair of exams that have common students. In a timetable a first order conflict (direct conflict) arises if a pair of conflicting exams is scheduled in the same timeslot. Second order conflicts arise from the conflicting exams that are schedule "too close" to each other but not in the same timeslot. In addition, a conflict graph can be created to indicate the conflict potentials among a set of exams. The *conflict graph* is an undirected graph with vertices being the exam set, and each pair of conflicting exams is connected by an edge.

Nearly in all variants of ETP problems, the conflict is considered. The first order conflict is forbidden so that no student is required to write two exams at the same time, which forms a hard constraint. Then, the most common soft constraint is to reduce second order conflicts, in other words, to spread conflicting exams properly so that students can have enough revision time between exams. However, in some relaxed cases, even the first order conflict is handled by soft constraints.

Thus, to handle the conflict, the essential information that we should include in the dataset of the problem is the students' enrollments for exams. For each exam we should have a list of students that will take it, not only the number of students. However, since the exam timetable usually should be published before students enroll for the exams, the enrollments are estimated, and the most common approach is using the students' enrollments for the courses. The problem model including students' enrollments is described as post-enrollment. Due to the complexity of this information, the traditional manual timetabling usually considers only the curriculum of degree programs, like in TU Darmstadt. Nevertheless, the post enrollment model is the most common and the most widely reported model for automated timetabling [McCollum et al., 2012]. So, in this thesis the post-enrollment model is employed. [Bergmann et al., 2014] described some defects of curriculum based model.

[Qu et al., 2009] summarized a list of the most common hard and soft constraints that occur in research, and we present it in Table 2.1 and 2.2. They can be roughly grouped as time related (No. 1 in Table 2.1 and Nos. 1-7 in Table 2.2) or resource related (No. 2 in Table 2.1 and Nos. 8-11 in Table 2.2). The constraints vary from one institution to another depending on their own rules, expectations and facilities.

Table 2.1: Primary hard constraints in ETP

	Primary hard constraints
1.	No first order conflict.
2.	Resources need to be sufficient (e.g., number of students below room capacity, enough rooms for all exams).

From the research papers we observe that the rules on room and time assignment especially vary a lot. An institution may have one of the following three room assignment rules:

1. Each exam is assigned to exactly one room.
2. Room-splitting: One exam can be split into several rooms.
3. Room-sharing: Several exams can share one room simultaneously.

Table 2.2: Primary soft constraints in ETP

	Primary soft constraints
1.	Reduce second order conflicts, i.e., spread exams as evenly as possible.
2.	Groups of exams required to take place at the same time, on the same day or at the same location.
3.	Exams to be consecutive, i.e., exams that should be scheduled back to back.
4.	Schedule all exams, or largest exams, as early as possible.
5.	An ordering (precedence) of exams needs to be satisfied.
6.	Limited number of students and/or exams in any timeslot.
7.	Time requirements (e.g. exams (not) to be in certain timeslots).
8.	Conflicting exams on the same day to be located nearby.
9.	Exam may be split over similar locations, when room splitting is allowed.
10.	Only exams of the same length can share the same room in the same timeslot, when room sharing is allowed.
11.	Resource requirements(e.g., room facility).

For rule 2 and 3, soft constraint 9 and 10 respectively need to be considered. Similarly, one of the following three time assignment rules will be employed:

1. Each exam is assigned to exactly one timeslot.
2. Timeslot-splitting: One exam can be split into several continuous timeslots.
3. Timeslot-sharing: Several exams can use the same room and timeslot as long as the sum of the exam durations is below the length of the timeslot, and then they will be scheduled sequentially manually.

A main challenge in ETP problem is the fact that exams have different durations. Rule 2 and 3 are advantageous in this case. They give more flexibility on the time so that a solution with higher quality may exist in search space. However, on the other hand, they highly enlarge the solution search space. Even though the splitting and sharing rules can be both employed in one problem, it usually does not happen in practice.

Since different institutions may have different and even contradictory rules, there are no universally accepted complete models so far.

2.2 Complexity

Many variants of ETP problems are proven to be NP-complete. By our definition, the ETP problem needs to do both time and room assignment, which forms a 3-index assignment problem. Some early papers in this field consider only time assignment as part of ETP problem. The basic problem is to assign exams to a limited number of timeslots with the constraints that there should be no first order conflict ([Carter and Laporte, 1996]). This basic problem can be easily mapped to a graph coloring problem, which is a well-known NP-complete problem, by considering timeslots as colors. Thus, this basic problem is easily proven to be NP-complete and the other variants are at least as difficult as the basic problem. However, in this thesis we did not provide a derivation of the complexity of our ETP problem.

2.3 Previous surveys and competitions

The first survey on examination timetabling was presented in [Carter, 1986]. It discussed many practical applications of exam timetabling algorithms, while none of those approaches was implemented in more than one institute. The survey was updated in [Carter and Laporte, 1996] which summarized the approaches used on exam timetabling between 1986 and 1996. The criteria for discussion of this paper, was that the method should be either tested on the real data or implemented in real-world applications.

To our knowledge, [Qu et al., 2009] is the newest survey on examination timetabling. On the basis of the paper [Carter and Laporte, 1996], [Qu et al., 2009] concentrated on the research between 1996 and 2009. The different approaches were classified and discussed. In addition, the paper presented a re-naming of the widely studied benchmark datasets to avoid a significant amount of confusion which had been a problem for many years. The paper also presented a range of potential future research directions and open issues in exam timetabling research.

Other than the surveys above, which concentrate on the exam timetabling, there are several surveys concerning the educational timetabling that we would like to recommend.

[Schaerf, 1999] looked at the formulations of school, course and exam timetabling and stated that the difference between the latter two are relatively small.

[Kristiansen and Stidsen, 2013] is a comprehensive survey on educational timetabling, which is concentrated on the four main education planning problems: University Course Timetabling, High School Timetabling, Examination Timetabling and Student Sectioning. It contains a section for exam timetabling, which is based on [Qu et al., 2009], and some new research papers are presented. It pointed out a difference between university course timetabling and exam timetabling: the former pursues a compact timetable whereas the latter pursues more spreading between events for each student.

The following two conferences are dedicated to the art of timetabling: the *International Conference on the Practice and Theory of Automated Timetabling* (PATAT) and the *Multidisciplinary International Scheduling Conference: Theory & Application* (MISTA). Both conferences are held every second year.

Besides the surveys and conferences mentioned above, there have been three *International Timetabling Competitions* (ITC) on educational timetabling problems respectively in 2003, 2007 and 2011. The second competition introduced three tracks along with associated benchmark datasets and one track is Examination Timetabling. To bridge the gap between theory and practice in exam timetabling, it presented a new model that better represents the complexity of the real-world situation in many institutions and the datasets are based on real-world instances as well. In this model, each exam uses exactly one timeslot and room-sharing rule is employed. The objective function is a weighted sum of several types of penalties, which are contributed by different soft constraints. The winner of the examination track of ITC2007 is [Müller, 2016]. More details of ITC2007 can be found on the website http://www.cs.qub.ac.uk/itc2007/.

In addition, the webpage http://www.cs.nott.ac.uk/~pszajp/timetabling/exam/ continuously collects together benchmark instances, solutions and software tools for examination timetabling.

2.4 Examination timetabling benchmark data

Due to the high interest in examination timetabling, various different benchmark problems have been established and widely studied. The following are some well-known benchmarks of exam timetabling, which are all named from the universities or competitions:

- Toronto benchmark data
- Nottingham benchmark data
- Melbourne benchmark data
- The second international timetabling competition (ITC2007) benchmark data
- Yeditepe benchmark data
- Purdue benchmark data

In this section, we only give a brief description of some of them. ITC2007 benchmark data has been briefly introduced in the previous subsection. [Qu et al., 2009] exhaustively described the former three benchmarks, clarified which papers dealt with which problems and summarized which of the methods that have appeared in the literature are the best on those benchmarks at that time. The links to all of the benchmark datasets can be found on http://www.cs.nott.ac.uk/~pszajp/timetabling/exam/.

2.4.1 Toronto benchmark data

[Carter et al., 1996] introduced a set of 13 real-world exam timetabling problems. The dataset of each problem merely contains the students' enrollments, i.e., a list of students for each exam. Since the original datasets are generic, many variants of these problems appeared in research by giving different objectives and additional time and resource settings (e.g., number of timeslots, maximal seating capacity per timeslot). [Qu et al., 2009] collected a complete list of all five variants that were studied in research papers and named them from Toronto a to Toronto e. In all variants only time assignment is considered. Furthermore, exam and timeslot durations are not considered and it simply assumes that each exam requires one timeslot. The constraint that there should be no first order conflict is included in all variants. In later variants, the maximal room capacity per timeslot (i.e., the maximal number of students in each timeslot) is considered. As for the objective, Toronto a is to minimize the number of timeslots needed for the problem, which is equivalent to a typical graph coloring problem. All the other variants have a aim to spread properly the conflicting exams, but have different evaluation functions.

In addition, [Qu et al., 2009] stated that five of the thirteen instances actually have two versions of the data that were circulated and were tested by different approaches during the years. They examined the data and found that in the second version of the datasets, duplicate exams have been assigned to some students in three instances. Thus, for these three instances they presented again a corrected second version of datasets. To avoid future confusion, they presented a re-naming of the datasets, and clarified which research papers have dealt with which dataset.

2.4.2 Purdue benchmark data

As we mentioned before, institutions may have different constraints, and therefore, [Müller, 2016] introduced a new benchmark of exam timetabling, which contains nine real-world datasets from Purdue University. The main differences between the ETP model in this benchmark and the ITC2007 ETP model are:

- Direct conflicts are allowed, but minimized.
- The room-splitting rule is employed instead of room-sharing.
- Rooms have two capacities, based on the seating mode.
- Penalties are calculated differently.

The benchmark datasets can be found on the website `http://www.unitime.org`.

2.5 Examination timetabling approaches

Generally, there are two types of methods: exact methods and heuristic methods. The exact methods concentrate on the formulation of a mathematical model for the ETP problem. Then, the model is encoded in the constraint programming, and solved by a commercial or open-source solver, or a well-known exact method for this mathematical model. The first part of this thesis focuses on an exact method. A MILP model is proposed and solved by a MILP solver. However, considering the size of real instances, exact methods usually needs a significant amount of computation time in both finding a good feasible solution and proving the optimality. Therefore, heuristic methods are demanded, which instead do not guarantee to reach the optimal solution.

Typically there are two types of heuristic methods. On the one hand, there are construction heuristics that produce a feasible solution from the scratch. On the other hand, there are improvement heuristics that try to improve a given feasible solution. ([Pochet and Wolsey, 2006])

Graph coloring heuristics are the construction heuristics that have been heavily studied in the early decades, as [Welsh and Powell, 1967] linked timetabling to graph coloring. The basic ETP problem of assigning timeslots to exams with the constraint that no conflicting exams are in the same timeslot, obviously corresponds to the vertex graph coloring on the conflict graph, which assigns colors to vertices so that no adjacent vertices have the same color. The basic graph coloring heuristics first order the exams by how difficult they are to be scheduled and then assign them, one by one, into the timeslot. The research focuses on the ordering strategies, and a broad range of them appear in the timetabling literature. [Qu et al., 2009] listed some of the widely used ordering strategies. Five of them were tested on the Toronto benchmark instances in [Carter et al., 1996].

Many metaheuristics are used in timetabling as improvement heuristics, and most widely used are local search (including Simulated Annealing, Tabu Search, Variable Neighborhood Search, Large Neighborhood Search, and etc.) and population-based metaheuristics (including Evolutionary Algorithms, Memetic Algorithms, Ant Algorithms, Artificial Immune Algorithms, and etc.). Especially population-based metaheuristics are attracting a high level of research attention in the recent decades. The effort required to tune the parameters of metaheuristics for specific problems to obtain high-quality solutions is usually very high, which is one significant drawback of metaheuristics.

In addition, the following techniques are integrated in many methods.

Since the timetabling needs to handle different constraints, multi-criteria techniques have been studied recently in timetabling. In the majority of approaches, weighted costs of violations of different constraints are summed and used to measure the quality of the solutions, which may have some deficiencies in real-world cases. Alternatively, multi-criteria techniques consider a vector of constraints instead of a single weighted sum.

Decomposition techniques are also employed in some methods. The idea of decomposition is that large problems are broken into small subproblems, for which optimal or high-quality solutions can be found by relatively simple techniques or in short time, since the search spaces of the subproblems are significantly smaller than that of the original problem ([Carter, 1983]). One way of decomposing the problems is to find the largest clique in the conflict graph. [Carter et al., 1996] integrated this technique into the graph coloring heuristics by scheduling the largest clique first. The reason is that, in a sense, the largest clique represents the most difficult group of exams to schedule since they are all in conflict. There are also other decomposition techniques, which are not be covered in this thesis.

One of the current trends in timetabling is the use of hybridization of different solution techniques. A common approach usually consists of a construction phase, realized by an exact or heuristic method, to find a feasible solution and a sequence of improvement heuristics to further improve it. [Merlot et al., 2002] employed constraint programming to generate initial solutions, and then a Simulated Annealing and a Hill Climbing method were used to improve the solutions. The winner of ITC2007 examination timetabling track [Müller, 2016] used an Iterative Forward Search algorithm for construction phase and then the Hill Climbing and Great Deluge to improve the solutions.

In the recent decades, hyper-heuristics are also attracting an increased level of research attention. Most of the heuristic methods are designed for one particular problem based on the knowledge of the search space. Such methods usually work poorly with other problems or even with another slightly different variant of problem. Hyper-heuristics are motivated by the goal of automating the design of heuristic methods to solve hard computational search problems ([Burke et al., 2013]). They can be described as heuristics to choose heuristics. That is, they operate on a search space of heuristics rather than directly on the search space of solutions. [Burke et al., 2013] did a comprehensive survey on the hyper-heuristics, and the reader can refer to it for more details.

2.5.1 MILP based approaches

In this subsection, we introduce several approaches in the literature of timetabling that employ the MILP model.

[Al-Yakoob et al., 2010] proposed a MILP model for the ETP problem in the Kuwait University (KU), which, to our knowledge, is the first MILP based approach in exam timetabling. Because the sizes of instances were relatively small, they were exactly solved by CPLEX (version 9.0), and the results are of significant improvements compared to the existing manual approach in KU.

[McCollum et al., 2012] proposed a MILP model for the ETP problem presented in ITC2007. In this model, the conflict graph is introduced to indicate the conflict potentials on exam pairs. The MILP model was also encoded in CPLEX Studio and the encoding can be found on the website http://www.cs.nott.ac.uk/~pszajp/timetabling/exam/. The author reported that it is not capable of solving ITC2007 instances.

In [Parkes and Ozcan, 2010] the same code with CPLEX 11 was used to exactly solve some of the smallest instances from Yeditepe University in Turkey and the author reported that the optimality has not been proven for any solutions of ETP instances in the Toronto and ITC2007 benchmarks.

Based on [McCollum et al., 2012], [Arbaoui et al., 2015] proposed new preprocessing stages and an improved MILP model so as to speed up the solving process. The preprocessing mainly concentrates on analyzing the data and then building a generalized conflict graph. Between two exams there is always the implicit competition for resources, which may also make it impossible to schedule two exams without common students simultaneously. Thus, a generalized conflict graph is introduced to indicate all exam pairs that cannot be scheduled simultaneously, not only those caused by the student conflicts. Besides, the MILP model was improved by adding some cuts.

[Bergmann et al., 2014] proposed a MILP model for the ETP problem in the Hamburg University of Technology (TUHH). In the model, room-splitting and timeslot-sharing rules are employed. It defines one timeslot to be 10 hours so that the complexity of problem is reduced. The Gurobi MIP solver was used to solve the real instances in TUHH. However, due to the size of instance, the best integer solution found by the solver within the time limit was not good enough. Thus it was further improved by a tabu-search based heuristic.

Based on a MILP formulation of ETP problem, [Bargetto et al., 2016] proposed a matheuristic solution approach that integrates a search phase realized by an exact algorithm (in this case the MILP solver) on a subset of the original problem into a well-known metaheuristic procedure. As metaheuristic, the Large Neighborhood Search (LNS) with multiple neighborhoods was employed. Based on the fundamental local search, LNS tries to reduce the risk of being trapped in a local optimum by using a very large neighborhood, and it still moves to the best solution in the neighborhood. In this approach, finding the optimum in the neighborhood is done by a MILP solver (CPLEX 12.6 is used in this paper). It is tested on the instance from Politecnico di Torino.

Based on the MILP model in [McCollum et al., 2012], the master thesis [Kadura, 2016] proposed a MILP model for the ETP problem in TU Darmstadt. Our MILP model is based on that, and a significant number of modifications and improvements have been made.

[Penn et al., 2017] proposed a MILP model for master operating theater timetabling in hospital and it is solved by FICO Xpress Optimization Suite. The problem is quite similar to exam timetabling.

Chapter 3

MILP model for ETP

In this chapter we formally introduce the ETP problem in TU Darmstadt and then propose a MILP model for it. To begin with, the ETP problem in TU Darmstadt is discussed and described verbally. Then, the parameters and the mathematical model composed of the variables, the objective function and the constraints for the optimization problem are presented. Our modeling is based on the MILP model proposed in the research paper [McCollum et al., 2012], which is for ETP problem in ITC2007, and the master thesis [Kadura, 2016] in TU Darmstadt. Some concepts and notations are retained from those papers. However, many individual parts of the model are significantly extended and modified, so that the model is suitable for ETP problem presented in this thesis. The main differences will be highlighted, however the reader should refer to [McCollum et al., 2012] for more details of the problem description and MILP model for ETP in ITC2007.

We use the following conventions of notations to help render the formulation more readable:

- Sets are upper case.
- Parameters are lower case.
- Variables are upper case.
- For matrices, a superscript is used to indicate the type of each dimension. For example, s^E and s^R are used for exam and room sizes respectively.

3.1 Problem description

3.1.1 General conditions and assumptions

In ETP problem, the three entities that we need to connect are exams, timeslots and rooms.

In this problem, timeslots have the same length and organized in two dimensions (inspired by [Penn et al., 2017]). The entire exam session is composed of a set of days $D = \{1, ..., n^D\}$, where n^D is the total number of days. Then, in one day the working hours, which should be continuous, (e.g., 10 hours from 8:00 to 18:00), are split into n^T continuous timeslots of equal duration (e.g., 2 hours). The index set of timeslots is denoted as $T = \{1, ..., n^T\}$. Thus, a timeslot is indicated by two indices day d and timeslot t.

Each room has a seating capacity and an availability matrix, which states during which timeslots the room can be used. In this problem, an exam is allowed to be split into several rooms. A group of rooms that can be used together by one exam is defined as a location(e.g., a campus or a building), and each room belongs to a location. Sometimes, it is not necessary to assign a specific room to an exam, instead, we can assign a room type and the specific room of that type can be chosen manually later. Thus, in oder to reduce the complexity, in this problem rooms are grouped into room types, and we classify rooms according to the seating capacity and location. Each room type has a seating capacity, location and availability matrix, which is derived from the availability of each room and states the number of available rooms of that type in each timeslot.

A pair of timeslot and room are defined as a resource block (as a pair of timeslot and frequency channel is defined as a resource block in 4G mobile communication). Thus, the ETP problem is to allocate a number of resource blocks to each exam. The room and time assignment rules in this problem are that room-sharing and timeslot-sharing are forbidden while room-splitting and timeslot-splitting are allowed. Hence, each resource block can only be used by one exam.

Thus, each exam has the following properties:

- Exam duration represented as a number of timeslots.
- Minimal and maximal number of rooms into which it can be split.
- Exam size: number of students enrolled in the exam.
- A list of students enrolled in the exam.

The time for handing out and collecting the papers has already been counted in the exam duration, so no interval is needed between two adjacent exams.

Since each exam can only use an integer number of timeslots, the timeslot length is somewhat crucial in the problem. A finer granularity gives more flexibility, but at the same time increases the problem complexity.

Our model concentrates on the student conflicts. Given the exam set E and students' enrollments, a conflict graph can be created. The conflict graph $G_C = (E, A^C)$ is an undirected graph with vertices being the set of exams E. The set of undirected edges A^C contains an edge $a = (i, j)$ if and only if exams i and j have a conflict potential, which means they have students in common. Since the edges are undirected, to prevent double counting and without loss of generality, we let A^C contain only arcs $a = (i, j)$ with $i < j$. A weight is associated to each edge $a = (i, j) \in A^C$ and in this thesis it is defined to be the number of common students. However, if desired this model can be extended by defining edge weights that depend also on other parameters, e.g., the credit points of the two exams.

The benefit of introducing the conflict graph is that it makes the model more generic, so that with slight modification this model can fit many other variants of ETP problems. There may be other reasons for forbidding two exams to be scheduled in the timeslot, not only the conflict of student. Such as, two exams need to use the same specified rooms, or two exams need the same proctor. To fit those cases, we only need to extend the conflict graph.

Even if a pair of exams has no conflict potential, it may still be impossible to schedule them simultaneously. Because, other than the conflict potential, between two exams there is also the implicit competition for resources.

In addition, the exams in TU Darmstadt are not jointly scheduled, instead, with some large exams preassigned by the university, each department schedules their own exams. Some resources, like rooms, are shared by all departments. In this thesis we consider the ETP problem in the department ETIT in TU Darmstadt.

The main differences on the conditions between our model and the model in [McCollum et al., 2012] is highlighted as follows:

- For room assignment, room-splitting is allowed in our model but no room-sharing.
- In [McCollum et al., 2012] the timeslots are not required to be of equal duration, and the timeslots on the same day are not required to be continuous. Then, the time assignment rule is that every exam should be assigned into exactly one timeslot with the constraint that the exam duration should not exceed the timeslot duration. As explained in the previous chapter, in the case that the exams have different durations, our time assignment rule may create better solution at the cost of the increase of the solution search space. To deal with the various types of exam duration, [Bergmann et al., 2014] used another strategy. They created timeslots of equal and very long duration, which should exceed the duration of any exam, and employed the timeslot-sharing rule. The reason why we did not employ that strategy is the fact that our rooms will not be available in such a long timeslot as many of them are shared with other departments
- Since every day has equal number of timeslots of equal duration, it becomes possible to organize the timeslots in a 2-dimensional matrix with one index indicating the day. The benefit is that it simplifies the implementation of the constraints concerning only the day of exams but not the exact time and reduces the complexity of model.

3.1.2 Hard constraints

The hard constraints considered in this problem are:

- Resource availability constraint: A room can be used only if it is available in that timeslot.
- Preassignment constraint: Some exams are preassigned to specific resource blocks or a group of resource blocks (e.g., a specific day or some specific timeslot in any day).
- Time hard constraint: Every exam must be allocated to the required number of continuous timeslots.
- Room split hard constraint: The number of rooms allocated to each exam must satisfy the demand.
- Location constraint: Every exam must be at exactly one location.
- Room capacity constraint: Exam size cannot exceed the total number of seats in the allocated rooms.

- Direct conflict constraint: Direct conflict is forbidden. No student should write two exams simultaneously.
- Exam coincidence constraint: Some exam pairs need to take place at the same time.

The time hard constraint and room split hard constraint jointly enforce another hard constraint: Complete allocation constraint, which is that all exams must take place.

In [McCollum et al., 2012], some hard constraints above are stated as RELAXABLE-HARD. In the case that the problem instance is infeasible, a relaxed problem can be created by allowing the RELAXABLE-HARD violations and minimizing them. A solution to the relaxed problem may help in identifying and repairing the sources of infeasibility. However, in this thesis, we do not study this relaxation and those constraints are always considered to be hard.

3.1.3 Soft constraints

As we discussed before, the violation of soft constraints is used to evaluate the quality of a feasible solution, so it contributes a penalty to the objective function. The objective function, which needs to be minimized, is a weighted sum of all types of penalties.

The penalties considered in this problem can be divided into two classes: *allocation penalty* and *conflict penalty*. The allocation penalties contributed by each exam depend only on its own room and/or timeslot allocation. However, the conflict penalties are caused by the second order conflicts, which depends only on the time gap between conflicting exams, but not the timeslot allocation. In [McCollum et al., 2012] the second class of penalty is referred as “pattern penalties" because we can also consider that they arise from restrictions on sequences of enrollment for each student.

The following allocation penalties are considered in this problem:

- **Time penalty:** A penalization on assigning an exam into a timeslot. For each exam, there is a different penalty associated with each timeslot. The overall time penalty is calculated by summing up such penalty caused by the time allocation of each exam.
- **Room penalty:** A penalization on assigning an exam into a room. Similarly, for each exam, there is a different penalty associated with each room type for using one room of that given type. The overall room penalty is calculated by summing up such penalty caused by the room allocation of each exam.
- **Room split penalty:** A penalization on assigning an exam into multiple rooms. Each exam can use one room without penalty, and every additional room will cause a penalty of one. In other words, the penalty of splitting an exam is equal to the number of additional rooms used by it.

The time and room penalties may consider the preference of proctors, the availability of resources and other potential factors. The constraint that exams with large size are preferred to be scheduled early is also considered in this problem. Instead of introducing another penalty, we encode it also in the time penalty, as it is indeed one type of preference on timeslots. In the experiments, the time and room penalties are actually not used, so in this thesis we do not specify how the penalty value should be defined. They are included in the model for completeness and potential future research.

The conflict penalties are penalization on the second order conflicts. The following conflicts are considered in this problem:

- **Two-in-a-row (back-to-back conflict):** Two conflicting exams are scheduled back to back on the same day, i.e., one exam begins immediately after the other ends.

- **Two-in-a-day:** Two conflicting exams are scheduled on the same day, including the case that they are back to back.

- **Exam spread:** The time gap between two conflicting exams, which counts only the number of days, is smaller than or equal to a specified minimal value. It includes also the case that they are on the same day, which means a gap of zero.

The penalty for one type of conflict counts the number of occurrences of that type of conflict, so each pair of conflicting exams will contribute a penalty equal to the number of common students if it have a conflict. Thus, the penalty can be calculated by traversing the conflict graph, and the penalty for one type of conflict is the sum of edge weight of the edges having that type of conflict.

In [McCollum et al., 2012] the conflict penalties are divided into exclusive types. For example, the two-in-a-day is defined as the case that two conflicting exams are scheduled on the same day but not back to back. Therefore, if two exams are scheduled back to back, the penalty will be counted only as part of the two-in-a-row penalty. However, in this thesis we do not use exclusive types. If two exams are scheduled back to back, the penalty will be counted three times. The advantage is that it reduces the complexity of model.

3.2 Sets and parameters

To provide a better overview of the notation used in the model, we present a list of all sets, indices and parameters as follows.

3.2.1 Sets

E	$= \{1, ..., n^E\}$ is the set of all exams, where n^E is the total number of exams.
E^S:	Set of exams to be scheduled. $E^S \subseteq E$. The rest of the exams in E will not be considered in the optimization.
E^P:	Set of exams that are preassigned to specific resource blocks or a group of resource blocks. $E^P \subseteq E^S$.
R	$= \{1, ..., n^R\}$ is the set of room types, where n^R is the total number of room types.
L	$= \{1, ..., n^L\}$ is the set of locations, where n^L is the total number of locations.
D	$= \{1, ..., n^D\}$ is the set of days, where n^D is the total number of days in the exam session.
T	$= \{1, ..., n^T\}$ is the set of timeslots in each day, where n^L is the total number of timeslots per day.
A^C:	Set of edges of the conflict graph with the exam set E^S, $G_C = (E^S, A^C)$. Since the edges are undirected, to prevent double counting and without loss of generality, we let A^C contain only arcs $a = (i, j)$ with $i < j$.
H^{coin}:	Set of pairs of exams (i, j) that must take place at the same time. For an exam pair (i, j) in this set, the order of i, j makes no difference.

In order to have a compact model, we will define variables and constraints only for the exam set E^S and the conflict graph is also defined only for E^S. However, the parameters related to the other exams are still retained in the model.

Instead of removing the exams that do not need to be scheduled and the parameters related to them, we keep them in the model for two reasons. First, a feasible instance should be able to provide resources required by all exams, not only the exams to be scheduled. The feasibility of the instance data will be checked before solving it and the details of feasibility testing will be explained in Section 3.6. Second, this option is required in the heuristic approach that will be introduced in the next chapter.

Besides, H^{coin} is not specified in the experiment instances, but it is included in the model for completeness.

3.2.2 Indices

e:	Indices for exams, $e \in E, E^S$ or E^P.
r:	Indices for room types, $r \in R$.
l:	Indices for locations, $l \in L$.
d:	Indices for days, $d \in D$.
t:	Indices for timeslots, $t \in T$.
(i, j):	A tuple of indices of two exams, used to represent an exam pair. $(i, j) \in H^{coin}$ or A^C.
a:	Indices for conflict edges, which are exam pairs with conflict potential, $a = (i, j) \in A^C$ where $i, j \in E^S$ and $i < j$.

3.2.3 Parameters

s_e^E: Size of exam $e \in E$, i.e., number of students that sit in exam e
d_e^E: Duration of exam $e \in E$, including time for preparation and follow up, represented as the number of timeslots.
b_e^E: Minimal number of rooms into which exam e can be split, $\forall e \in E$.
m_e^E: Maximal number of rooms into which exam e can be split, $\forall e \in E$.
u_r^R: Number of available rooms of type $r \in R$.
s_r^R: Seating capacity of a room belonging to room type $r \in R$, i.e., number of seats in the room.
v_{rl}^{RL}: Indicates the location of room. Equal to 1 if room type r is in location l, 0 otherwise, $\forall r \in R, l \in L$.
u_{rdt}^{RDT}: Resource availability. It is the number of available rooms of type r in timeslot t on day d, $\forall r \in R, d \in D, t \in T$. It is introduced because the availability of each room may change over time.
p_{erdt}^{ERDT}: Preassignment. It is the number of rooms of type r in timeslot t on day d pressigned to exam e, $\forall e \in E^P, r \in R, d \in D, t \in T$.

The weights and parameters for soft constraint penalties:

w_{edt}^T: A weight that specifies the penalty for exam e using timeslots t on day d, $\forall e \in E, t \in T, d \in D$.
w_{er}^R: A weight that specifies the penalty for exam e using one room of type r, $\forall e \in E, r \in R$.
w^{RS}: Weight for room split penalty.
w^{2R}: Weight for two-in-a-row (back-to-back conflict) penalty.
w^{2D}: Weight for two-in-a-day penalty.
w^{ES}: Weight for exam spread penalty.
g: The preferred minimal gap in days between two exams with conflict potential. An exam pair with conflict potential that is scheduled closer than g days contributes to the exam spread penalty.
w_a^C: Weight for each edge $a = (i, j) \in A^C$. In this thesis, it is defined to be the number of students taking both of the two exams i and j.

3.3 Variables

3.3.1 Primary decision variables

The primary integer decision variables are:

$$X_{er}^{ER} = \text{number of rooms of type } r \text{ assigned to exam } e, \qquad \forall e \in E^S, r \in R \quad (3.1)$$

$$B_{edt}^{EDT} = \begin{cases} 1, & \text{if exam } e \text{ begins in timeslot } t \text{ on day } d, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall e \in E^S, d \in D, t \in T \quad (3.2)$$

Obviously, those variables respectively give the information which room type and how many rooms are assigned to an exam and which timeslot is assigned to it as the beginning timeslot. An unique assignment is fixed when those variables are determined. X^{ER}_{er} takes non-negative integer values, while B^{EDT}_{edt} is binary.

3.3.2 Secondary variables

The term secondary variables means that the values of those variables will be directly forced given any legal assignment to the primary variables. These variables are used to write the constraints and to compute the objective function.

In order to encode the direct conflict constraints and compute the time penalty, we will use the variables:

$$X^{EDT}_{edt} = \begin{cases} 1, & \text{if exam } e \text{ is in timeslot } t \text{ on day } d, \\ 0, & \text{otherwise.} \end{cases}$$
$$\forall e \in E^S, d \in D, t \in T \quad (3.3)$$

When the assignment is fixed, the value of those variables can be determined from the beginning time of the exams B^{EDT}_{edt} and the duration of the exams d^E_e. If B^{EDT}_{edt} is one for exam e in timeslot t on day d, X^{EDT}_{edt} must be forced to be one for all d^E_e successive timeslots from t. It will be enforced by constraints (3.28).

In order to encode the resource availability constraints and preassignment constraints, we will use the variables:

$$X^{ERDT}_{erdt} = \text{number of rooms of type } r \text{ in timeslot } t \text{ on day } d \text{ allocated to exam } e,$$
$$\forall e \in E^S, r \in R \quad (3.4)$$

and their values determined by

$$X^{ERDT}_{erdt} \equiv X^{ER}_{er} X^{EDT}_{edt}, \ \forall e \in E^S, r \in R, d \in D, t \in T \quad (3.5)$$

In order to encode the location constraints, the following variables are introduced to indicate the location of each exam.

$$X^{EL}_{el} = \begin{cases} 1, & \text{if exam } e \text{ is in location } l, \\ 0, & \text{otherwise.} \end{cases}$$
$$\forall e \in E^S, l \in L \quad (3.6)$$

Their values are determined from X^{ER}_{er} by the rule: if at least one room in location l is allocated to exam e, then X^{EL}_{el} is forced to be 1.

The penalties for violations of the various soft constraints are encoded as non-negative variables as follows:

C^T: Time penalty
C^R: Room penalty
C^{RS}: Room split penalty
C^{2R}: Two-in-a-row (back-to-back conflict) penalty
C^{2D}: Two-in-a-day penalty
C^{ES}: Exam spread penalty

Because the last three penalties depend on the relative position of all exam pairs with conflict potential in the schedule, we can compute those penalties by summing the penalty component caused by each conflict edge. Thus, we again define the binary variables C_a^{2R}, C_a^{2D} and C_a^{ES} to indicate whether edge $a = (i, j) \in A^C$ incurs a penalty.

$$C_a^{2R} = \begin{cases} 1, & \text{if edge } a = (i, j) \text{ incurs a two-in-a-row penalty,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\forall a = (i, j) \in A^C \quad (3.7)$$

and the same rule applies to C_a^{2D} and C_a^{ES}.

All those relations will be formulated as linear constraints.

We also indicate that the six variables for the six types of penalty are not strictly necessary, because their values will be directly computed from the values of the other variables. However, before solving the model CPLEX will simplify it by removing the redundancy, so those penalty variables can remain in our model in order to clearly indicate the sources of the penalty. Also for the same reason, they can be relaxed to be real values if desired, while the rest of the secondary variables should be forced to be integer or binary.

3.4 Objective

Minimize

$$C^T + C^R + w^{RS}C^{RS} + w^{2R}C^{2R} + w^{2D}C^{2D} + w^{ES}C^{ES} \quad (3.8)$$

The objective function measures the quality of the solutions. It is the weighted sum of all penalties, which are caused by the violation of various types of soft constraints. As in [McCollum et al., 2012], there are no additional weights for the time and room penalties C^T and C^R since the associated weights were already included in their definitions. As we discussed previously this problem is multi-objective, but the weighted sum approach is employed in this model for simplicity. By giving different weights, institutions can lay emphasis on different penalties.

3.5 Constraints

3.5.1 Hard constraints

The hard constraints are presented as follows.

The range constrains for decision variables:

$$X_{erdt}^{ERDT}, X_{er}^{ER} \geq 0 \quad (3.9)$$

$$X_{el}^{EL}, X_{edt}^{EDT}, B_{edt}^{EDT} \in \{0,1\} \quad (3.10)$$

$$C_a^{2R}, C_a^{2D}, C_a^{ES} \geq 0 \quad (3.11)$$

$$C^T, C^R, C^{RS}, C^{2R}, C^{2D}, C^{ES} \geq 0 \quad (3.12)$$

The secondary variables for the penalties, $C^T, C^R, C^{RS}, C^{2R}, C^{2D}, C^{ES}$, can be relaxed to be real values if desired, since their values will be directly computed from the values of the other variables. The rest of the variables can only take integer or binary values.

Resource availability constraints:

$$\sum_{e \in E^S} X_{erdt}^{ERDT} \leq u_{rdt}^{RDT}, \qquad \forall r \in R, d \in D, t \in T \tag{3.13}$$

$$X_{er}^{ER} \leq u_r^R, \qquad \forall e \in E^S, r \in R \tag{3.14}$$

The total number of used rooms of type r in any timeslot and the number of rooms used by any exam should be limited by the number of available rooms.

Preassignment constraints:

$$X_{erdt}^{ERDT} \leq p_{erdt}^{ERDT}, \qquad \forall e \in E^P, r \in R, d \in D, t \in T \tag{3.15}$$

Exams in the set E^P can only be assigned to preassigned resource blocks.

Time hard constraints:

$$\sum_{d \in D} \sum_{t \in T} B_{edt}^{EDT} = 1, \qquad \forall e \in E^S \tag{3.16}$$

$$\sum_{d \in D} \sum_{t \in T} X_{edt}^{EDT} = d_e^E, \qquad \forall e \in E^S \tag{3.17}$$

Every exam must begin exactly once and be allocated to the required number of timeslots. Constraint (3.28) will guarantee that the allocated timeslots are continuous.

Room split hard constraints:

$$\sum_{r \in R} X_{er}^{ER} \geq b_e^E, \qquad \forall e \in E^S \tag{3.18}$$

$$\sum_{r \in R} X_{er}^{ER} \leq m_e^E, \qquad \forall e \in E^S \tag{3.19}$$

Every exam e must be allocated to at least b_e^E rooms and at most m_e^E rooms.

Location constraints:

$$\sum_{l \in L} X_{el}^{EL} = 1, \qquad \forall e \in E^S \tag{3.20}$$

Every exam must be allocated to exactly one location.

Room capacity constraints:

$$s_e^E \leq \sum_{r \in R} s_r^R X_{er}^{ER}, \qquad \forall e \in E^S \tag{3.21}$$

The rooms allocated to each exam should provide a sufficient total seating capacity.

Direct conflict constraints:

$$X_{idt}^{EDT} + X_{jdt}^{EDT} \leq 1, \qquad \forall a = (i,j) \in A^C, d \in D, t \in T \tag{3.22}$$

Direct conflict is not allowed. Every exam pair with conflict potential cannot take place at the same time.

Exam coincidence constraints:

$$X_{idt}^{EDT} = X_{jdt}^{EDT}, \qquad \forall (i,j) \in H^{coin}, d \in D, t \in T \tag{3.23}$$

Every exam pair in H^{coin} must be scheduled simultaneously.

The following constraints link the secondary decision variables to the primary decision variables:

$$\sum_{d\in D}\sum_{t\in T} X^{ERDT}_{erdt} = d^E_e X^{ER}_{er}, \qquad \forall e \in E^S, r \in R \tag{3.24}$$

$$X^{ERDT}_{erdt} \le X^{ER}_{er}, \qquad \forall e \in E^S, r \in R, d \in D, t \in T \tag{3.25}$$

$$X^{ER}_{er} v^{RL}_{rl} \le u^R_r X^{EL}_{el}, \qquad \forall e \in E^S, r \in R, l \in L \tag{3.26}$$

$$X^{ERDT}_{erdt} \le u^{RDT}_{rdt} X^{EDT}_{edt}, \qquad \forall e \in E^S, r \in R, d \in D, t \in T \tag{3.27}$$

$$X^{EDT}_{edt} = \sum_{t'=\max\{t-(d^E_e-1),1\}}^{t} B^{EDT}_{edt'}, \qquad \forall e \in E^S, d \in D, t \in T \tag{3.28}$$

Constraint (3.28) ensures that d^E_e continuous timeslots will be allocated to exam e. If B^{EDT}_{edt} is one for exam e in timeslot t on day d, X^{EDT}_{edt} must be forced to be one for all d^E_e successive timeslots from t, i.e., from t to $t + (d^E_e - 1)$. An equivalent approach is that, X^{EDT}_{edt} must be forced to be one if B^{EDT}_{edt} has a value of one for any one of the d^E_e timeslots before t, i.e., from $t - (d^E_e - 1)$ to t. Note that it should count only the part of timeslots in set T if $t - (d^E_i - 1) < 1$, so the value X^{EDT}_{edt} can be obtained by the sum of $B^{EDT}_{edt'}$ in the range $t' = \{\max\{t - (d^E_e - 1), 1\}, ..., t\}$.

In MIP, *cuts* or *cutting planes* are constraints that reduce the feasible region for the LP relaxation but not the feasible region of the original problem. They cut away non-integer solutions that would otherwise be solutions of the continuous relaxation. The addition of cuts usually reduces the number of branches needed in the branch-and-bound approach to solve a MIP and, therefore, speed up the solving process. Thus, the following cuts are included in the model. The following constraints enforce that a exam cannot be scheduled to begin in a timeslot if there are no enough successive timeslots on that day.

$$B^{EDT}_{edt} = 0, \qquad \forall e \in E^S, d \in D, t \in T, t \ge n^T - d^E_e \tag{3.29}$$

3.5.2 Soft constraints

Allocation penalties

The overall time penalty is calculated by summing up the time penalty caused by the time allocation of each exam:

$$C^T = \sum_{e\in E^S}\sum_{d\in D}\sum_{t\in T} w^T_{edt} X^{EDT}_{edt}, \tag{3.30}$$

as allocating exam e into timeslot t on day d causes a penalty of w^T_{edt} in the time penalty and X^{EDT}_{edt} indicates the time allocation.

Similarly, the overall room penalty is calculated by summing up the room penalty caused by the room allocation of each exam:

$$C^R = \sum_{e\in E^S}\sum_{r\in R} w^R_{er} X^{ER}_{er}, \tag{3.31}$$

as allocating exam e into one room of type r causes a penalty of w^R_{er} in the room penalty and X^{ER}_{er} indicates the room allocation.

Room split penalty:

$$C^{RS} = \sum_{e \in E^S} \left(\sum_{r \in R} X^{ER}_{er} - 1 \right) \tag{3.32}$$

It is the total number of additional rooms.

Conflict penalties

The overall two-in-a-row penalty is the sum of violations on each edge weighted by the edge weight, i.e., the number of common students:

$$C^{2R} = \sum_{a \in A^C} w^C_a C^{2R}_a \tag{3.33}$$

The minimization of C^{2R} within the overall objective will automatically force C^{2R}_a towards zero. Then the following constraints forces it to be one when necessary:

$$B^{EDT}_{idt} + B^{EDT}_{jd(t+d^E_i)} \leq 1 + C^{2R}_a \tag{3.34a}$$

$$B^{EDT}_{id(t+d^E_j)} + B^{EDT}_{jdt} \leq 1 + C^{2R}_a \tag{3.34b}$$

$$\forall a = (i,j) \in A^C, d \in D, t \in \{1, ..., (n^T - d^E_i - d^E_j + 1)\}$$

Those constraints state that C^{2R}_a will be forced to be one if exam j directly follows exam i or it happens in the inverse order.

Similarly, the following constraints compute the two-in-a-day penalty and exam spread penalty.

Two-in-a-day penalty:

$$C^{2D} = \sum_{a \in A^C} w^C_a C^{2D}_a \tag{3.35}$$

$$\begin{aligned} &\sum_{t \in T} \left(B^{EDT}_{idt} + B^{EDT}_{jdt} \right) \leq 1 + C^{2D}_a \\ &\forall a = (i,j) \in A^C, d \in D \end{aligned} \tag{3.36}$$

C^{2D}_a is forced to be one if exams i and j are on the same day; otherwise, the minimization of C^{2D} forces it to be zero.

Exam spread penalty:

$$C^{ES} = \sum_{a \in A^C} w^C_a C^{ES}_a \tag{3.37}$$

$$\begin{aligned} &\sum_{d'=d}^{d+g} \left(X^{ED}_{id'} + X^{ED}_{jd'} \right) \leq 1 + C^{ES}_a \\ &\forall a = (i,j) \in A^C, d \in \{1..(n^D - g)\} \end{aligned} \tag{3.38}$$

C^{ES}_a is forced to be one if the gap between exams i and j is smaller than or equal to g days; otherwise, the minimization of C^{ES} forces it to be zero.

Penalty lower bound

From constraint (3.32) we can see that the room split penalty has an obvious lower bound:

$$C^{RS} \geq \sum_{e \in E^S} \left(b_e^E - 1 \right) \tag{3.39}$$

since b_e^E is the minimal number of rooms assigned to exam e. Thus, the room split penalty contributes a lower bound to the objective function:

$$w^{RS} \sum_{e \in E^S} \left(b_e^E - 1 \right) \tag{3.40}$$

3.6 Instance feasibility testing

Given the complexity of the problem, it is possible that the user may specify a problem instance that is infeasible. Although for some instances CPLEX identifies the infeasibility in a short time, it is not always true. In order to avoid wasting time in solving an infeasible instance, the data feasibility should be tested before optimization. In most cases, the infeasibility is caused by the insufficiency of resources, so in this thesis only the following conditions for checking the sufficiency of resources are used.

$$u_{rdt}^{RDT} \leq u_r^R, \ \forall r \in R, d \in D, t \in T \tag{3.41}$$

$$d_e^E b_e^E \leq \sum_{r \in R} \sum_{d \in D} \sum_{t \in T} p_{erdt}^{ERDT}, \ \forall e \in E^P \tag{3.42}$$

$$\sum_{e \in E} d_e^E b_e^E \leq \sum_{r \in R} \sum_{d \in D} \sum_{t \in T} u_{rdt}^{RDT} \tag{3.43}$$

Conditions (3.41) state that the number of available rooms of type r at any time cannot be larger than the total number of available rooms of type r. Conditions (3.42) state that the number of resource blocks preassigned to exam e cannot be smaller than the minimal number required by exam e. Conditions (3.43) state that the total number of available resource blocks cannot be smaller than the minimal number required by all exams in E, not only exams to be scheduled.

The modeling language OPL provides "assertions" to verify the consistency of the model data. The data feasibility testing can be implemented with this function.

Chapter 4

Hierarchical Construction Approach

After having established the MILP model for ETP problem, we can encode it in CPLEX Studio and then solve it with CPLEX. However, as we will show in Chapter 5, the real instances cannot be exactly solved in reasonable time, and neither a high-quality solution can be found. As is known, optimizing a MIP model involves two aims:

1. Finding a better integer feasible solution.

2. Improving the best bound so as to prove that there is no better solution undiscovered.

For our instances, both two directions consume a significant amount of workload.

Our key aim is to achieve high-quality solutions within a short computational time, so a heuristic approach is required. Many early heuristics were based on a simulation of the human way of solving the problem [Schaerf, 1999]. When scheduling exams manually, the administrators usually schedule them one by one until all exams have been scheduled. Thus, an intuitive approach is to divide the exams into several subsets and schedule them sequentially. Besides, if only a small part of the entire problem is considered, then the exact MILP approach is still considerable. Therefore, we would like to find a construction heuristic approach that consists of a sequence of search phases realized by an exact algorithm (MILP solver) on a subset of the original problem. In summary, we propose the *hierarchical construction approach* that decomposes the entire ETP problem into a hierarchy of subproblems. The subproblems are still ETP problems but each of them schedules only a subset of the exams so that for each subproblem an optimal or high-quality solution can be found by the MILP solver in short time. Then, the solution to the entire ETP problem can be obtained by solving the sequence of subproblems using the exact MILP approach.

Since the time and room penalties are actually not specified in the experiment instances, the conflict penalties contribute to the major part of the objective function. Besides, the exam coincidence constraints (3.23) and the preassignment are not used in the entire ETP problem. Thus, our heuristic approach concentrates on dealing with the conflicts: eliminating first order conflicts and reducing second order conflicts. The decomposition of the problem then corresponds to the decomposition of the conflict graph.

The decomposition techniques, based on the conflict graph, used to create the hierarchy will be introduced in Section 4.1. In Section 4.2 we will improve the MILP model by adding some cuts, and in Section 4.3 the hierarchical construction algorithm will be completely illustrated and several issues will be discussed.

4.1 Decomposition techniques

The basic idea of the hierarchical construction approach is to decompose the total exam set into multiple hierarchical layers and then sequentially schedule each layer by solving an ETP subproblem formed by the layer. Except for the first layer, each layer is composed of the entire previous layer, which has already been scheduled and will be fixed using the preassignment constraints in the subproblem, and the other exams selected to schedule in this layer. Therefore, the subproblem considers only the edges (conflict potentials) in the layer.

We first introduce the methods to create the first layer, then with a slight modification we obtain the method to create the other lower layers.

4.1.1 The first layer

A principal of forming the hierarchy is that the most “constrained” or “important” exams should be scheduled first. By “important” we mean that the assignment of the exam has a heavy effect on the solution quality.

It is reasonable to jointly schedule a set of exams in which most of the exam pairs have conflict potential, which corresponds to a dense subgraph in the conflict graph. In an undirected graph, a clique is a complete subgraph in which every pair of vertices is connected by an edge in graph. In the conflict graph of ETP a clique corresponds to a set of mutually conflicting exams. The *maximum clique* is the one with maximum cardinality. In conclusion, a reasonable choice of the first layer is the maximum clique as they, in a sense, represents the most difficult group of exams to schedule.

The *Maximum Clique Problem* (MCP) is to find a clique of the largest size in a given graph. It is a well-known NP-hard problem and can also be formulated as a MILP problem. Many exact algorithms for solving the Maximum Clique Problem have been proposed. Most of them employ some form of branch-and-bound approach, which is generally a regular approach for MIP problem, combined with some novel pruning techniques tunned to this problem, like in [Pattabiraman et al., 2013]. However, since we are already using CPLEX solver for MIP problem, we will also model this problem with OPL and then solve it with CPLEX. The time for finding maximum clique with our instances is actually negligible.

For simplicity, the MCP problem is formulated directly with the undirected conflict graph $G_C = (E, A^C)$ with the total exam set E, but the reader should be aware that this model is valid for any undirected graph. The complement graph of $G_C = (E, A^C)$ is the graph $\bar{G_C} = (E, \bar{A^C})$, where $\bar{A^C} = \{(i,j)|i,j \in E, i < j, (i,j) \notin A^C\}$, i.e., it contains all node pairs not directly connected by an edge. Note that, as we mentioned in the previous chapter, in order to prevent double counting for undirected edges and without loss of generality, we consider only node pairs (i,j) with $i < j$.

An equivalent interpretation of the MCP problem is to find the largest subset of vertices in which every pair of vertices is connected. Thus, the problem can be formulated as

$$\text{maximize} \sum_{i \in E} x_i \tag{4.1}$$

$$\text{s.t.} \quad x_i + x_j \leq 1, \forall (i,j) \in \bar{A^C} \tag{4.2}$$

$$x_i \in \{0,1\}, \forall i \in E \tag{4.3}$$

where the binary variable x_i is used to indicate whether node i is selected into the set. The constraints enforce that if the node pair is not directly connected, then they should not be both selected.

In the experiments, MCP is actually not used to form the first layer, but the introduction of MCP will help the reader to easily understand the derivation of the following approaches.

As the conflict penalties considers the edge weights, the edge weights should be considered when we form the first layer. Therefore, the *maximum weight clique* would be a more reasonable choice.

The *Maximum Weight Clique Problem* (MWCP) is to find a clique of the largest weight in the graph. A graph can have both node weights and edge weights, which causes two variants of MWCP problem. One defines the clique weight as the sum of weights of all nodes in the clique, and the other defines it as the sum of edge weights. In this thesis the term MWCP problem always indicates the second variant. By slightly modifying the model for MCP, we can obtain the model for the MWCP problem:

$$\text{maximize} \sum_{(i,j) \in A^C} w_{ij}^C y_{ij} \tag{4.4}$$

$$\text{s.t.} \quad x_i + x_j \leq 1, \forall (i,j) \in \bar{A^C} \tag{4.5}$$

$$x_i \geq y_{ij}, x_j \geq y_{ij}, \forall (i,j) \in A^C \tag{4.6}$$

$$x_i + x_j \leq 1 + y_{ij}, \forall (i,j) \in A^C \tag{4.7}$$

$$x_i \in \{0,1\}, \forall i \in E \tag{4.8}$$

$$y_{ij} \in \{0,1\}, \forall (i,j) \in A^C \tag{4.9}$$

where the additional binary variable y_{ij} is introduced for computing the weight of the subgraph, and it indicates whether the edge (i,j) is selected into the subgraph. Obviously, an edge is selected *iff* the two nodes of this edge are selected.

$$y_{ij} = 1 \Leftrightarrow x_i = 1 \text{ and } x_j = 1, \qquad \forall (i,j) \in A^C \tag{4.10}$$

Thus, the additional constraints (4.6) and (4.7) are introduced to link variables y_{ij} to x_i and x_j. The constraints (4.6) enforce that x_i and x_j are forced to be one if y_{ij} is one, and the constraints (4.7) enforce that y_{ij} is forced to be one if both x_i and x_j are one.

The time for solving MWCP with our experiment instances is still negligible.

In the case that the maximum weight clique contains relatively few exams, another approach is needed so that the first layer can have a proper size. In order to select a larger subgraph, we should consider not only the cliques but also other subgraphs that are sufficiently dense.

The term "*quasi-clique*" is used to describe a sufficiently dense subgraph. In this thesis, we provide two definitions of quasi-clique and one of them is stricter than the other.

First, a quasi-clique with a specified minimal density d is defined as a subgraph whose density is not less than d. For a graph, the graph density is defined as the ratio between the number of edges in the graph and the maximal number of edges. Thus, assuming the size of the (undirected) subgraph is n, an equivalent definition is that the number of edges cannot be less than $\frac{1}{2}dn(n-1)$. This definition is referred to as definition 1. Similarly, we will find the *maximum weight quasi-clique*, with a specified minimal density d, to form the first layer. The problem is referred to as *Maximum Weight Quasi-Clique Problem* (MWQCP1, 1 indicates quasi-clique definition 1) and formulated as:

$$\text{maximize} \sum_{(i,j)\in A^C} w_{ij}^C y_{ij} \tag{4.11}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A^C} y_{ij} \geq \frac{1}{2} d \left(\sum_{k\in E} x_k \right) \left(\sum_{l\in E} x_l - 1 \right) \tag{4.12}$$

$$x_i \geq y_{ij}, x_j \geq y_{ij}, \forall (i,j) \in A^C \tag{4.13}$$

$$x_i + x_j \leq 1 + y_{ij}, \forall (i,j) \in A^C \tag{4.14}$$

$$x_i \in \{0,1\}, \forall i \in E \tag{4.15}$$

$$y_{ij} \in \{0,1\}, \forall (i,j) \in A^C \tag{4.16}$$

Constraint (4.12) constrains the number of edges in the subgraph, whose size is derived from the sum of variables x_i. MWQCP1 is not linear, instead it is a MIP problem with convex quadratic constraints, which can also be solved by CPLEX.

Then, we used also a stricter definition, which constrains the number of edges of each node, not simply the total number of edges of the subgraph. By definition, every node in a clique of size n is connected to all other $n-1$ nodes within the clique. Thus, a quasi-clique of minimal density d with n nodes is defined as a subgraph where each node is connected to at least $d(n-1)$ nodes. It is denoted as definition 2, and the Maximum Weight Quasi-Clique Problem with definition 2 (MWQCP2) is formulated as:

$$\text{maximize} \sum_{(i,j)\in A^C} w_{ij}^C y_{ij} \tag{4.17}$$

$$\text{s.t.} \quad d \left(\sum_{k\in E} x_k \right) - \left(\sum_{(i,j)\in A^C} y_{ij} + \sum_{(j,i)\in A^C} y_{ji} \right) \leq M(1 - x_i), \quad \forall i \in E \tag{4.18}$$

$$x_i \geq y_{ij}, x_j \geq y_{ij}, \forall (i,j) \in A^C \tag{4.19}$$

$$x_i + x_j \leq 1 + y_{ij}, \forall (i,j) \in A^C \tag{4.20}$$

$$x_i \in \{0,1\}, \forall i \in E \tag{4.21}$$

$$y_{ij} \in \{0,1\}, \forall (i,j) \in A^C \tag{4.22}$$

M is a upper bound of the left-hand side of the inequality and it can be chosen as $d(n^E - 1)$, where n^E is the cardinality of E as introduced in the previous chapter.

The parameter d has the same meaning, the minimal density of the subgraph, in both two definitions. However, the number of edges can only be integer. Thus, the minimal number of edges in definition 1 is $\lceil \frac{1}{2}dn(n-1) \rceil$, where $\lceil a \rceil$ denotes the smallest integer $\geq a$ $(a > 0)$. In definition 2, the minimal number of edges of each node is $\lceil d(n-1) \rceil$ and, hence, the minimal number of edges of the subgraph is $\lceil \frac{1}{2}n\lceil d(n-1) \rceil \rceil \geq \lceil \frac{1}{2}dn(n-1) \rceil$. In both two definitions and with any density, the set of quasi-cliques includes the set of cliques. Both two definitions of quasi-clique are generalization of the concept of clique, and a clique can be considered as a quasi-clique (in both two definitions) with a density of 1.

As we can see from the definitions, with the same minimal density d, the set of quasi-cliques in definition 1 completely includes the set of quasi-cliques in definition 2, and both includes the set of cliques. Therefore, MWQCP1 is a relaxation of MWQCP2, and MWQCP2 is a relaxation of MWCP. Due to the relaxation, the time for solving MWQCP1 and MWQCP2 increases a lot, and, in fact, in the experiment the optimum was not found within the time limit for both two problems. Thus, the best solution found within the specified time limit will be used to form the first layer.

In addition, the following two constraints are added into all the models introduced above:

$$x_i = 1, \forall i \in E^C \tag{4.23}$$

$$x_i = 0, \forall i \in E^F \tag{4.24}$$

Obviously, those constraints restrict the feasible set of the problems by specifying which nodes must be selected and which are forbidden to be selected. Now, the problems are generalized to search for maximum weight clique or quasi-clique in the set of subgraphs that contain all nodes in E^C and does not contain any node in E^F. Those generalized problems are introduced to create the other lower layers, and the complete procedures will be introduced in the next subsection.

Furthermore, as MWQCP1 and MWQCP2 cannot be exactly solved in short time, we propose again an alternative approach to form the first layer. With those additional constraints we can further make a restriction of MWQCP1 and MWQCP2 by specifying E^C to be the maximum weight clique, because the maximum weight quasi-clique very likely contains the maximum weight clique. The restriction is referred to as *MWC restriction*. The restricted problems can be solved within a moderate time for our experiment instances. Thus, this approach contains two stages. First, we need to solve the MWCP problem, whose solving time is negligible. Then, knowing the maximum weight clique, the restricted MWQCP1 or MWQCP2 problem is solved to eventually form the first layer.

In conclusion, in this subsection we proposed three methods (MWCP, MWQCP1 and MWQCP2) to select a set of exams to form the first layer of the multi-layer heuristic, and in all methods the selection of exams is formulated to a MIP problem and then solved by CPLEX. When the quasi-clique methods are used, a density of the first layer needs to be specified. Besides, the MWC restriction can be added into the quasi-clique methods in order to reduce the computation time.

In the experiments, the proposed construction methods of layer 1: MWCP and unrestricted and restricted MWQCP1 and MWQCP2 with several values of minimal density were respectively used, and the results will be presented and compared in Chapter 5.

4.1.2 The other lower layers

After having scheduled one layer, we need to select other unscheduled exams, together with the scheduled exams, to form the next layer. As we said before, in this ETP subproblem the assignment of the scheduled exams will be fixed using the preassignment constraints in the MILP model and the other unscheduled exams are selected to jointly schedule in this layer. Since we are concentrating on dealing with the conflicts, it is straightforward that we should schedule first the exams connected to the subgraph of scheduled exams in the conflict graph. As well, to schedule one exam, we should jointly consider the other exams that form a dense subgraph together with it.

Therefore, given the scheduled layer, the procedure to form the next layer is presented in Algorithm 1.

Algorithm 1 Layer construction

INITIALIZATION: Given the conflict graph and the scheduled layer;

1. We first find in the conflict graph the neighborhood of the subgraph of scheduled exams. The neighborhood of a subgraph is defined as the set of all nodes that are not in the subgraph and are connected to at least one node in the subgraph.

2. - If the neighborhood is empty (in the rare case that the conflict graph is disconnected), we find the maximum weight clique in the subgraph of unscheduled exams.
   - Otherwise, for each exam in the neighborhood, we find in the subgraph of unscheduled exams the maximum weight clique containing that exam. It is done by formulating a generalized MWCP problem with E^C being that exam and E^F being the set of scheduled exams.

   Finally, all these cliques are selected to form the next layer together with the scheduled exams.

In the construction of the lower layers, only the MWCP approach is used, while relaxing it to be a quasi-clique is not considered.

4.2 Improved MILP model with clique cuts

As explained in the previous chapter, cuts are useful for reducing the computation time. In [Arbaoui et al., 2015] the clique cuts were added into their improved MILP model. Since some cliques will be found in the phase of selecting exams for layers in our heuristic approach, it is better to improve the MILP model by including the clique cuts:

$$\sum_{e \in c} X_{edt}^{EDT} \leq 1, \qquad \forall d \in D, t \in T, c \in \mathcal{C} \tag{4.25}$$

where c is a clique, which is a set of exams, and $\mathcal{C}$ is a set of cliques. This constraint simply enforces that in any timeslot at most one of all exams in a clique can take place as they are all connected to each other in the conflict graph. This constraint is a cut for the MILP problem, since the direct conflict constraints (3.22) already enforce that there should be no direct conflict.

4.3 Hierarchical construction algorithm

Using the decomposition techniques discussed previously, we decompose the total exam set into multiple hierarchical layers. In each iteration, given a partial timetable and a new layer created by the previous iteration, an ETP subproblem is formulated with the MILP model. The exam set to be scheduled E^S is the current layer. Among them, the exams in the previous layer have already been scheduled and, hence, are preassigned according to the partial solution. The exams not in the layer will not be considered. Then, the subproblem is solved by CPLEX, and depending on the result we will prepare the data for the next iteration. If an integer feasible solution is found then we update the partial timetable and select other unscheduled exams to form the next layer. Otherwise, if the subproblem is found to be infeasible, we have to go back to the previous layer and extend the layer to include all unscheduled exams. Thus, the timetable is extended, layer by layer, until all exams have been scheduled.

The complete hierarchical construction algorithm is presented in Algorithm 2. As mentioned before, in the experiments the following layer 1 construction methods will be used: MWCP and unrestricted and restricted MWQCP1 and MWQCP2 with several values of minimal density. This algorithm is also implemented in CPLEX Studio.

In general, there are two drawbacks in decomposition techniques ([Qu et al., 2009]). First, early assignments may lead to later infeasibility, which is actually also a problem encountered in other constructive methods. Thus, in our heuristic algorithm the backtracking to previous layer is added to handle this drawback. Second, globally high-quality solutions may be missed as certain soft constraints cannot be evaluated when the problems are decomposed, which will be discussed in the following.

The substance of this hierarchical construction approach is that it explores only a part of the search space of the entire ETP problem that most likely contains high-quality solutions. Since it avoids the exhaustive exploration of the search space, it can find a high-quality solution in short time, but does not guarantee to reach the optimal solution. Obviously, the choice of the part to be explored plays an important role in this approach, and it directly controls the trade-off between the computation time and the quality of the final solution.

The sizes of the layers certainly control the size of the part of search space to be explored, and, hence, have a crucial effect on the performance of this approach. If we define small layers, the number of layers is large and the subproblems can be easily solved. However, the part of the search space that we explored is relatively small. In contrast, if we use large layers, the number of layers is small and the time for solving each subproblem will be very long. Thus, the sizes of the layers must be well tuned.

Besides, A subproblem may still not be small enough to be exactly solved in short time, so in that case the best integer solution found within a specified time limit is used. For simplicity, we give the same time limit to each subproblem. However, it brings an issue that the complexity of subproblems should be balanced. Moreover, it is straightforward that different values of time limit need to work with different hierarchies to produce a good solution. The time limit and the maximal number of layers are used to limit the overall computation time.

In addition, parameter tuning can also be an issue, since it plays a significant role in the approach. When the hierarchical construction approach is used, the following parameters need to be specified:

- The construction method of layer 1, three options: MWCP, MWQCP1 and MWQCP2.
- When quasi-clique methods are used, we need to specify the minimal density and whether the MWC restriction is employed.
- The time limit for solving each subproblem.
- The maximal number of layers is usually irrelevant, but it is included in the approach for completeness.

The parameters, other than the time limit for subproblems, together control the hierarchy, and then the performance of this approach directly depends on the hierarchy and the time limit for subproblems.

Besides, in the case that no subproblem is infeasible during the scheduling, the hierarchy can actually be determined before scheduling.

Algorithm 2 Hierarchical Construction Algorithm

INITIALIZATION:

- Give the dataset of the entire ETP problem.
- Define the layer 1 construction method, the maximal number of layers K_{max} and time limit for ETP subproblem J_{max}.
- Initialize the set of scheduled exams to be empty.

1. Create layer 1 using the defined method, and prepare the dataset for the ETP subproblem in layer 1 and save it in a file. E^S is set to be the exam set of layer 1 and $E^P = \emptyset$ in layer 1. Define the layer number $k = 1$.
2. Formulate the ETP subproblem in layer k and solve it using CPLEX with time limit J_{max}. If an integer feasible solution is found within J_{max}, go to step 3; otherwise, go to step 4.
3. Update the sets of scheduled exams to be the exam set of layer k. Next,
    - If all exams have been scheduled, report the solution and exit.
    - Otherwise, create layer $k+1$:
        * If $k < K_{max} - 1$, then create layer $k+1$ using Algorithm 1.
        * Otherwise, set layer $k+1$ to be the total exam set E.

    Prepare the dataset of layer $k+1$. E^S is set to be the exam set of layer $k+1$, E^P is set to be the exam set of layer k and they are preassigned according to the partial solution. Set $k = k+1$ and go to step 2.
4. - If we are in layer 1, report that no solution has been found for the entire ETP problem and exit.
    - Otherwise, we have to go back to previous layer. Modify dataset of layer $k-1$ by extending layer $k-1$ to be the total exam set E. Update the set of scheduled exams correspondingly. Set $k = k-1$ and go to step 2.

Chapter 5

Experiments and Results

Tests were done on one login node of Lichtenberg High Performance Computer (Lichtenberg HPC) in TU Darmstadt, which has 4 Intel Xeon E5-4650 (Sandy Bridge, AVX) processors (= 4x8 = 32 CPU cores) with 2.7 GHz and 128 GB RAM. Since the computer is shared by many users, we consistently monitored the processes during our process running so that we guaranteed that CPU load consumed by other processes were negligible.

The software IBM ILOG CPLEX Optimization Studio (CPLEX Studio) of version 12.8.0 is used for encoding the MIP model and implementing the heuristic algorithm. The modeling language used in CPLEX Studio is IBM ILOG Optimization Programming Language (OPL), and a script language IBM ILOG Script is also provided for controlling the optimizations. CPLEX Studio provides two solvers: CPLEX and CP. The CPLEX solver is used in this thesis. The reader is recommended to refer to the CPLEX documentation provided by IBM for more details of the software, OPL and CPLEX solver. The CPLEX solver worked with the following configuration for all MIP problems:

- MIP emphasis: integer feasibility.
- MIP search method: dynamic search + heuristic.
- Parallel mode: deterministic, using up to 32 threads.

Dynamic search is similar to the traditional branch-and-cut method but implemented in a different way, and the integrated heuristic is employed periodically to find an potential integer solution for the MIP problem. Tuning CPLEX parameters is not a part of this thesis, so it was just the default configuration except for the MIP emphasis. In the experiments we used several values for the time limit. Besides, the data of the search tree is set to be compressed and stored on disk, so the size of the search tree can be considered as unlimited.

The test instances is introduced in Section 5.1, and then the results on the instances respectively using the exact MILP approach and the hierarchical construction approach are presented and discussed in Section 5.2.

5.1 Instances

Unfortunately we did not get a dataset of students' enrollments in ETIT TU Darmstadt. Therefore, we used one of the Toronto benchmark datasets, named as ear83IIc in [Qu et al., 2009], because it contains only the students' enrollments and the problem size is close to our real scenario. The characteristics of the Toronto ear83IIc is shown in Table 5.1, where the conflict density is the density of the conflict graph.

Table 5.1: Characteristics of the Toronto ear83IIc

Exams	Students	Enrollments	Min Ex Size
189	1108	8057	1
Max Ex Size	Edges	Conflict density	
232	4849	0.27	

Based on that, we add other parameters to make a complete dataset of our problem. We made two instances, named respectively as ear83IIc-1 and ear83IIc-2, which have different room configurations. [Parkes and Ozcan, 2010] assigned a maximal capacity per timeslot of 350. So, in configuration 1 we created 5 rooms with that total capacity. Configuration 1 has 5 room types with different capacities, each of which has only one room, and it considers only one location. Room configuration 1 is shown in Table 5.2. Configuration 2 is the real scenario in TU Darmstadt, and it has 19 room types located in 2 campuses. The total capacity per timeslot is 1609. Room configuration 2 is shown in Table 5.3.

Table 5.2: Room configuration 1

Capacity	150	100	50	30	20
No. Rooms	1	1	1	1	1

In both instances, the exam session has 20 days (4 weeks with 5 days per week). Each day has 10 hours and is divided into 5 timeslots of 2 hours.

Since the Toronto benchmark did not consider exam durations, each exam is randomly assigned a duration of 1 or 2 timeslots (i.e., 2 or 4 hours, which is common in TU Darmstadt) with equal probability. The minimal and maximal number of rooms into which each exam can be split is respectively set to be 1 and 4. However, if exam size exceeds capacity of any room, then its minimal number of rooms is set to be 2.

All rooms are available in any timeslot. Weights in the objective function for different penalties are all set to be 1 and the preferred minimal gap g is set to be 1 day. The exam coincidence constraint (3.23) and the time and room penalties are actually not used. The instances have no preassigned exams.

Table 5.3: Room configuration 2

Capacity	Location	No. Rooms
123	1	1
80	1	2
62	1	1
49	1	2
42	1	2
36	1	4
27	1	5
25	1	3
20	1	6
14	1	5
12	1	3
113	2	1
55	2	1
49	2	1
45	2	1
33	2	3
20	2	2
17	2	4
11	2	3

5.2 Results

5.2.1 Exact MILP approach

We first present the results with the exact MILP approach. The clique cuts proposed in Chapter 4 are also included in the MILP model, but only the maximum weight clique is considered. Table 5.4 shows the results on instance ear83IIc-1 with a 2-hour time limit, including the penalty of the best integer solution (UB) and the best lower bound (LB). The gap is defined as the relative distance between them:

$$\text{gap} := \frac{\text{UB - LB}}{\text{UB}} \tag{5.1}$$

There are 8 exams larger than the largest room. Therefore, the lower bound of 8 is contributed by the room split penalty, and CPLEX solver did not find a tighter lower bound. The gap is very large because both the upper bound and the lower bound are very poor. Later, we extended the time limit to 8 hours, but neither the upper bound nor the lower bound was improved. Since ear83II-c has a higher complexity, CPLEX solver did not find any integer solution for it within the 8-hour time limit.

5.2.2 Hierarchical construction approach

We tested the hierarchical construction approach on both two instances with several configurations of layer 1 construction. All three construction methods of layer 1: MWCP,

Table 5.4: Results on instance ear83IIc-1 in exact MILP approach

Best Integer	Best Bound	Gap
4023	8	99.80%

MWQCP1 and MWQCP2 were tested. For quasi-clique methods, two values of density, 0.9 and 0.8, were used, and both restricted and unrestricted cases were tested. We also tested several values of time limit for solving subproblems, and the maximal number of layers was set to be a large number so that it would be irrelevant. In all test cases, the infeasibility of subproblems did not happen, so the backtracking was actually not used.

To begin with, we observe the hierarchy built with different configurations, since it directly controls the final result. in Table 5.5 we present the results of the construction of layer 1 with different configurations. There are totally nine configurations, which differ on the construction method of layer 1, the minimal density in MWQCP, and whether the MWC restriction is used in MWQCP. Since the decomposition of the problem does not depend on the room configuration, both two instances have the same results, and, hence, in the table the test instance is denoted as ear83IIc. The time limit for solving the problem of construction of layer 1 was set to be 10 minutes, only the unrestricted MWQCP1 and MWQCP2 did not reach the optimal solution no matter which value of density was used.

As we should expect, when we make a relaxation from clique to quasi-clique or from a higher minimal density to a lower minimal density, we obtain a subgraph with higher (or at least equal) weight. When the minimal density is set to be 0.8, for both MWQCP1 and MWQCP2, the restricted problem has the same solution as the unrestricted. However, since the unrestricted problem did not reach the optimal solution, no conclusion can be made about the optimality. Because our decomposition is deterministic, starting from the same layer 1 we would certainly build the same hierarchy, which, in consequence, leads to the same timetable. Thus, in fact only seven hierarchies were built. In Table 5.6, we present and number the seven hierarchies built with different configurations. In all cases, the problem was decomposed into three layers. Besides, even though the size of layer 1 varies from 21 to 54, layer 2 and, in consequence, layer 3 do not change much.

Table 5.5: Results of layer 1 construction on ear83IIc with different configurations

Layer 1 Construction Method	MWCP	MWQCP1				MWQCP2			
Layer 1 Density	1	0.9		0.8		0.9		0.8	
MWC Restriction	-	Yes	No	Yes	No	Yes	No	Yes	No
Optimal Size Weight	Yes 21 3322	Yes 38 7281	No 32 6240	Yes 54 10392	No 54 10392	Yes 27 4170	No 25 4906	Yes 37 7167	No 37 7167

Then, the results on the two instances using the hierarchical construction approach with the seven hierarchies are presented respectively in Table 5.7 and 5.8. Two values of time

Table 5.6: Hierarchies on ear83IIc with different configurations

Layer 1 Construction Method	MWCP	MWQCP1			MWQCP2		
Layer 1 Density	1	0.9		0.8	0.9		0.8
MWC Restriction	-	Yes	No	-	Yes	No	-
Hierarchy No.	1	2	3	4	5	6	7
Layer 1 Size	21	38	32	54	27	25	37
Layer 2 Size	179	182	182	182	179	181	182
Layer 3 Size	189	189	189	189	189	189	189

limit have been tested: 10 minutes and 1 hour. The tables indicate the value of objective function of final solutions with each hierarchy and each value of time limit. The hierarchies are ordered by the size of layer 1. Besides, in all test cases, the subproblems in first two layers did not reach the optimal solution, and only the problem in layer 3, which schedules only fewer than 10 exams, was solved in seconds.

First of all, compared with the exact approach, this heuristic approach improves the solutions in all test cases, even in a shorter time.

We first focus on the results on instance ear83IIc-1. As we stated before, the performance of our hierarchical construction approach depends on both the hierarchy and the time limit for subproblems and, hence, different values of time limit need to work with different hierarchies to provide the best performance. For instance ear83IIc-1, the best hierarchy working with the 10-minute time limit is the third, while the best with the 1-hour time limit is the second. Since in all hierarchies the layer 2 and 3 are almost fixed, layer 1 defines the number of exams to be scheduled respectively in layer 1 and layer 2. As we mentioned before, the best performance is achieved when the sizes of those two layers are balanced, and the performance is worse if anyone of them is too larger than the other. The results are compatible with this discussion. When the time limit is 10 minutes, with increasing the size of layer 1, the solutions first are improved and then get worse. Another observation is that, a hierarchy with a smaller layer 1 has a better performance in the short run (short time for solving problem), but a worse performance in the long run. In addition, increasing the time for solving subproblems always improves the solutions for all hierarchies, and moreover, the difference on the quality of solution between hierarchies is reduced. When the time limit is 1 hour, all hierarchies, except for the fourth, have almost equally good performance.

Instance ear83IIc-2 has more resources (rooms) but, in consequence, higher complexity. When the time limit is 10 minutes, it is hard to find any pattern in the results, and the sixth hierarchy has the best performance. When the time limit is 1 hour, similarly, all hierarchies, except for the first (MWCP), have almost equally good performance, and the second hierarchy has the best performance. Even though this instance has a higher complexity compared to the first instance, our heuristic approach still finds the solutions of almost the same level of quality.

Another thing we observe in the experiments is that, in all test cases, the subproblem in layer 3, which schedules only fewer than 10 exams, is solved in seconds, and the exams are scheduled without increasing the penalties. This suggests that, as we wanted, our hierarchical construction approach puts the exams easy to schedule into lower layers.

In addition, from the results we can also see that tuning the parameters in this approach for a specific instance may really require a significant amount of effort.

Table 5.7: Results on ear83IIc-1 in hierarchical construction approach with different hierarchies and time limits

Hierarchy No.	1	6	5	3	7	2	4
Layer 1 Size	21	25	27	32	37	38	54
Time Limit 600s	3204	2631	2475	1497	1603	1738	2160
Time Limit 3600s	1497	1530	1419	1455	1523	1396	2096

Table 5.8: Results on ear83IIc-2 in hierarchical construction approach with different hierarchies and time limits

Hierarchy No.	1	6	5	3	7	2	4
Layer 1 Size	21	25	27	32	37	38	54
Time Limit 600s	2995	1964	3759	1994	2260	2930	2525
Time Limit 3600s	2389	1509	1572	1488	1570	1423	1514

Because the test instances are not exactly real-world, we do not have results with manual timetabling. Besides, it is also difficult to estimate a tight lower bound for our ETP problem. Therefore, we can only compare the results of this heuristic approach with that of the exact MILP approach. The conclusion is that, compared to the exact MILP approach, the hierarchical construction approach heavily improves the solutions.

Chapter 6

Conclusion and Future Work

In this thesis, we have presented the ETP problem in TU Darmstadt and proposed a MILP model for it. In order to achieve high-quality solutions within a short computation time, we have proposed the hierarchical construction approach. This approach decomposes the entire ETP problem into a hierarchy of subproblems based on the conflict graph, so that each subproblem can be easily handled using exact MILP approach. This approach concentrates on dealing with the conflicts. The results of test on some instances show that, compared with the exact MILP approach, the hierarchical construction approach considerably improves the solutions to the entire problem even in shorter time.

However, there are still many issues that might be considered in the future research.

First, it is difficult to evaluate the performance of our hierarchical construction approach, since there are neither good solutions obtained by other approaches nor a tight lower bound. Therefore, we should also apply our approach on the ITC2007 benchmark problems, for which there is a significant number of good solutions found by other approaches.

Besides, some other constraints, e.g., preassignment, exam coincidence, time and room penalty, have not been considered in our approach yet. They should be included in the future research.

Another potential research direction is to improve the construction method for the hierarchy. For example, we can also try different construction methods for the lower layers.

Furthermore, the performance of our hierarchical construction approach may be potentially improved by using the generalized conflict graph. As we introduced in Chapter 2, the generalized conflict graph considers also the incompatibilities between exams caused by the competition for resources.

Additionally, for the moment, we give the same time limit to each subproblem, which brings an issue that the complexity of subproblems should be balanced. However, the hierarchy also depends on the structure of the conflict graph, so in some cases it may be impossible to create a balanced hierarchy. Alternatively, we can use variable time limits, while one drawback is that more parameters need to be tuned.

Parameter tuning is also one challenge in this approach and it needs a high level of attention in the future research. The machine learning techniques may potentially be used for the parameter tuning, like in [Battistutta et al., 2017].

Bibliography

[Al-Yakoob et al., 2010] Al-Yakoob, S. M., Sherali, H. D., and Al-Jazzaf, M. (2010). A mixed-integer mathematical modeling approach to exam timetabling. *Computational Management Science*, 7(1):19.

[Arbaoui et al., 2015] Arbaoui, T., Boufflet, J.-P., and Moukrim, A. (2015). Preprocessing and an improved mip model for examination timetabling. *Annals of Operations Research*, 229(1):19–40.

[Bargetto et al., 2016] Bargetto, R., Della Croce, F., and Salassa, F. (2016). A matheuristic approach for an examination scheduling problem. In *Proceedings of the 11th International Conference on the Practice and Theory of Automated Timetabling*, pages 467–471.

[Battistutta et al., 2017] Battistutta, M., Schaerf, A., and Urli, T. (2017). Feature-based tuning of single-stage simulated annealing for examination timetabling. *Annals of Operations Research*, 252(2):239–254.

[Bergmann et al., 2014] Bergmann, L. K., Fischer, K., and Zurheide, S. (2014). A linear mixed-integer model for realistic examination timetabling problems. In *Proceedings of the 10th International Conference on the Practice and Theory of Automated Timetabling*, pages 82–101.

[Burke et al., 1995] Burke, E., Elliman, D., Ford, P., and Weare, R. (1995). Examination timetabling in british universities: A survey. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 76–90. Springer.

[Burke et al., 2013] Burke, E. K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., and Qu, R. (2013). Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724.

[Carter, 1983] Carter, M. (1983). *A Decomposition Algorithm for Practical Timetabling Problems*. Department of Industrial Engineering, University of Toronto.

[Carter, 1986] Carter, M. W. (1986). Or practice—a survey of practical applications of examination timetabling algorithms. *Operations research*, 34(2):193–202.

[Carter and Johnson, 2001] Carter, M. W. and Johnson, D. G. (2001). Extended clique initialisation in examination timetabling. *Journal of the Operational Research Society*, 52(5):538–544.

[Carter and Laporte, 1996] Carter, M. W. and Laporte, G. (1996). Recent developments in practical examination timetabling. In Burke, E. and Ross, P., editors, *Practice and Theory of Automated Timetabling*, pages 1–21, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Carter et al., 1996] Carter, M. W., Laporte, G., and Lee, S. Y. (1996). Examination timetabling: Algorithmic strategies and applications. *Journal of the Operational Research Society*, 47(3):373–383.

[Kadura, 2016] Kadura, A. (2016). Modellierung der klausurterminplanung am beispiel der tu darmstadt.

[Kristiansen and Stidsen, 2013] Kristiansen, S. and Stidsen, T. R. (2013). A comprehensive study of educational timetabling, a survey. *Department of Management Engineering, Technical University of Denmark.(DTU Management Engineering Report*, (8.2013).

[McCollum et al., 2012] McCollum, B., McMullan, P., Parkes, A. J., Burke, E. K., and Qu, R. (2012). A new model for automated examination timetabling. *Annals of Operations Research*, 194(1):291–315.

[Merlot et al., 2002] Merlot, L. T., Boland, N., Hughes, B. D., and Stuckey, P. J. (2002). A hybrid algorithm for the examination timetabling problem. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 207–231. Springer.

[Müller, 2016] Müller, T. (2016). Real-life examination timetabling. *Journal of Scheduling*, 19(3):257–270.

[Pardalos and Xue, 1994] Pardalos, P. M. and Xue, J. (1994). The maximum clique problem. *Journal of global Optimization*, 4(3):301–328.

[Parkes and Ozcan, 2010] Parkes, A. J. and Ozcan, E. (2010). Properties of yeditepe examination timetabling benchmark instances. In *Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling*, pages 531–534.

[Pattabiraman et al., 2013] Pattabiraman, B., Patwary, M. M. A., Gebremedhin, A. H., Liao, W.-k., and Choudhary, A. (2013). Fast algorithms for the maximum clique problem on massive sparse graphs. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 156–169. Springer.

[Penn et al., 2017] Penn, M., Potts, C. N., and Harper, P. R. (2017). Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables. *European Journal of Operational Research*, 262(1):194–206.

[Pochet and Wolsey, 2006] Pochet, Y. and Wolsey, L. A. (2006). *Production planning by mixed integer programming*. Springer Science & Business Media.

[Qu et al., 2009] Qu, R., Burke, E. K., McCollum, B., Merlot, L. T., and Lee, S. Y. (2009). A survey of search methodologies and automated system development for examination timetabling. *Journal of scheduling*, 12(1):55–89.

[Schaerf, 1999] Schaerf, A. (1999). A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127.

[Welsh and Powell, 1967] Welsh, D. J. A. and Powell, M. B. (1967). An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10(1):85–86.