A Decision Support Tool for Scheduling Master´s Thesis Defences

Lara Emília Jaco Ramos

Dissertation to obtain the Master of Science Degree in

Industrial Engineering and Management

Supervisor(s): Prof. Daniel Rebelo dos Santos
Prof. José Rui de Matos Figueira

Examination Committee

Chairperson: Prof. António Sérgio Constantino Folgado Ribeiro
Supervisor(s): Prof. Daniel Rebelo dos Santos
Members of the Committee: Prof. Miguel Alves Pereira

June 2023

Acknowledgements

The conclusion of this dissertation represents the end of a stage, the conclusion of my academic journey, for which I could not fail to thank all those who helped, motivated and accompanied me throughout all these years.

To my supervisors, Daniel Santos and José Rui Figueira, my special thank you for all the knowledge transmitted and for all the availability always demonstrated throughout the realization of my dissertation to discuss ideas and clarify certain indecisions. Thank you for guiding and motivating me throughout this process, and for all the advice and concern shown.

I would like to express my deep gratitude to my role-models, mother, father, sister and grandmother, for all their support, not only at an academic level, but throughout my life. Thank you for teaching me that nothing is achieved without effort. I would also like to thank my friend who has always been with me, Maria Amiga, who even far away advised and motivated me with the right words at the right time.

And as Técnico is not done alone, I could not fail to thank all the friends who have been with me since day one. Mapeia, Leonor and Mariana, for being present in all the memories I have of these years, without which I would not have been able to achieve my goals, Luís, Nuno and Tomás for all the support and laughter that made this journey lighter and Maria Roomie for always being one door away to hear me.

To all of you who have been part of this journey, my most sincere thank you.

Resumo

Os problemas de calendarização educacionais são problemas com uma grande presença na literatura, contando com um vasto número de estudos realizados neste sentido, devido à grande complexidade destes problemas. Este trabalho centrou a sua investigação no caso de estudo do Departamento de Engenharia e Gestão, do Instituto Superior Técnico, Universidade de Lisboa, Portugal. Atualmente, a alocação das defesas de tese é realizada de forma manual, demonstrando ser pouco eficiente e uma sobrecarga de trabalho para a secretária do departamento devido à elevada complexidade. O problema é modelado através de programação linear inteira mista, com o objetivo de minimizar o número de dias em que cada membro tem de assistir a defesas de tese e o número de *slots* vazios entre defesas alocadas em cada dia, para um determinado membro. Um modelo capaz de definir a constituição de cada júri simultaneamente à alocação de cada defesa de tese é posteriormente definido. Ambos os modelos são testados com instâncias geradas computacionalmente. O primeiro modelo não consegue encontrar soluções viáveis para todos as instâncias, lacuna colmatada com o segundo modelo, devido à flexibilidade de alocação dos membros do jurí que permite a obtenção de melhores horários. Por sua vez, o segundo modelo não apresenta soluções viáveis dentro do tempo computacional limite, para instâncias de maior dimensão. Os dois modelos demonstraram ser uma boa solução para a otimização do processo realizado manualmente no departamento, sendo necessária a escolha entre horários mais compactos ou menor tempo computacional.

Palavras-Chave: Problema de Calendarização Educacional, Problema de Calendarização de Defesa de Tese, Modelo Matemático Exato, Programação Linear Inteira

Abstract

Educational scheduling problems are problems inserted in the context of schools and universities, with a large presence in the literature, with a vast number of researches carried out in this sense, due to the great complexity of the problems. This study focused its investigation on the case study of the Department of Engineering and Management, Instituto Superior Técnico, University of Lisbon, Portugal. Currently, the allocation is carried out manually, proving to be inefficient and an overload of work for the department secretary due to the high complexity. The problem is modeled through mixed integer linear programming, with the objective of minimizing the number of days that each member has to attend thesis defences and the number of empty slots between defences allocated on each day, for a given member. A model capable of defining the constitution of each thesis defence jury simultaneously with its allocation is also defined. Both models are tested with computationally generated instances. The first model fails to find feasible solutions for all instances, a gap filled with the second model, due to the flexibility of allocation of jury members which allows obtaining better schedules. In turn, the second model fails to present feasible solutions within the computational time limit, for larger instances. The models proved to be a good solution for optimizing the process performed manually in the department, requiring the choice between more compact schedules or less computational time.

Keywords: Educational Timetabling Problem, Thesis Defence Timetabling Problem, Exact Mathematical Model, Integer Linear Programming

Contents

List of Figures

List of Tables

Acronyms

ETP	Education Timetabling Problems
STP	School Timetabling Problems
UTP	University Timetabling Problems
ITC	International Timetabling Competition
UCTP	University Course Timetabling Problems
UETP	University Examination Timetabling Problem
TDTP	Thesis Defense Timetabling Problem
MEGI	Master’s in Industrial Engineering and Management
IST	Instituto Superior Técnico
DEG	Department of Engineering and Management

1. Introduction

Chapter 1 of the dissertation presents the topic and purpose of the research. Section 1.1 gives background information about the problem being investigated. Section 1.2 outlines the goals and aims of the research. Finally, in section 1.3 the overall structure of the dissertation is described.

1.1 Overview

The definition of timetabling problems involves the allocation of a specific resource to a particular timeslot while ensuring that the constraints defined are adhered to as closely as possible, and the overall objective of the task is achieved (Wren, 1995 [144]). Timetabling problems encompass a diverse range of areas such as train scheduling, sports events, employee shifts, and academic class scheduling (Cacchiani & Paolo, 2012 [28], Trick, 2000 [137], Meisels & Schaerf, 2003 [83] and Pillay, 2016 [97]). The investigation into timetabling problems dates back several decades, with the initial research by Gotlieb, in 1963 [58], followed by numerous articles on the same topic. The research has led to the development of higher quality and diverse models, with an emphasis on automatic scheduling solutions as highlighted in Schaerf's work, in 1999 [119].

Despite the extended history of research into timetabling problems in educational contexts, the investigation of thesis defence allocation problems is relatively new. The first study on this topic was conducted by Huynh et al., in 2012 [61] and represents the initial exploration of the problem in the literature. Since this initial study, several other investigations have contributed to the literature on thesis defence allocation problems. However, due to the recency of this topic, there is still ample opportunity for further research to advance our understanding of this area. There is a particular opportunity to apply and analyze models constructed for educational problems in the allocation of thesis defences, further contributing to the development of effective solutions for this problem.

The allocation of thesis defences is a common challenge faced by universities that confer master's degrees. Typically, this problem is addressed manually by collecting the availability of each jury member and identifying a suitable timeslot where everyone can witness the defence. This responsibility often falls on a single individual, resulting in a significant workload and a complex problem, particularly when managing a large number of theses to be allocated. Additionally, the distribution of thesis defences over multiple days is often suboptimal, with no consideration given to compact and optimized allocation.

In the context of this dissertation, the Department of Engineering and Management (DEG) at Instituto Superior Técnico (IST) serves as a case study. At DEG, the scheduling of thesis defences is assigned to a single person, the department's secretary. This process is highly time-consuming and ineffective, as it requires the secretary to individually contact all members of the jury to collect their availability. In a subsequent manual phase, the secretary must verify which timeslots are suitable for the thesis defence based on the availability of the jury members. At times, it may be impossible to find a suitable timeslot where all members are available, requiring additional availability collection and resulting in a further increase in workload. This process must be carried out twice per school year, further adding to the secretary's workload.

In an effort to streamline and optimize the process of scheduling thesis defences, this dissertation proposes a mathematical model that takes into account the preferences of committee members in order to create more efficient and higher quality schedules. Specifically, the proposed mathematical models evaluate different ways of scheduling the thesis defences, from which recommendations are produced. Additionally, this optimization is particularly relevant to DEG, as the university has two separate campi and better allocation of thesis defences would minimize the need for unnecessary travel.

1.2 Research Objectives

The primary aim of this dissertation is to develop mathematical models capable of evaluating different ways of scheduling thesis defences, to optimize the allocation for students pursuing the Master's degree in Industrial Engineering and Management (MEGI). These models test potential modifications to the current manual process and ensure a more efficient allocation of defences, taking into account the preferences of professors, the committee members, and resulting in better quality, more compact schedules. To achieve this objective, the following specific goals need to be fulfilled:

- Describe the relevant regulations for thesis defence allocation in IST, including deadlines and committee composition requirements;
- Provide an overview of the MEGI program;
- Characterize the current thesis defence allocation process in DEG, including identifying the person responsible and how the process is carried out;
- Conduct a literature review on timetabling problems, with a focus on Educational Timetabling Problems (ETP);
- Analyze various ETP methods and assess the potential adaptations for the Thesis defence Timetabling Problems (TDTP);
- Develop exact mathematical models that can solve the TDTP for DEG, incorporating specific

constraints of the department and variations in the way that committee composition is elaborated;

- Create instances that replicate the case study under consideration;
- Utilize the instances to test the models and compare the results obtained between them, as well as with the current manual process. The evaluation criteria will include the time taken to obtain a solution, the feasibility of the solutions, and the values of the objective function, which reflect the quality of the schedules generated;
- Provide recommendations and insights for DEG at IST, identifying possible areas for future research.

1.3 Document Structure

This dissertation consists of 7 chapters:

- **Chapter 1 - Introduction**

  The purpose of this chapter is to provide context for the problem, giving an overview of it, highlighting and detailing the research objectives, and outlining the structure of the document.

- **Chapter 2 - Problem Definition**

  This chapter provides context for the case study examined in the dissertation. It begins by outlining the regulations at the IST that pertain to thesis defences, with a specific focus on the MEGI program. The chapter then proceeds to describe the current process used to allocate thesis defences.

- **Chapter 3 - Literature Review**

  This chapter presents a literature review focused on timetabling problems, with a particular emphasis on ETP. ETP includes thesis defence allocation problems as well as other types of scheduling problems that arise in schools and universities.

- **Chapter 4 - Mathematical Model**

  Chapter 4 presents the construction of a mathematical model based on the current manual process of allocating thesis defences in the department under study. The chapter presents the indices, parameters, and decision variables used in the model, as well as the constraints considered and the objective function defined.

- **Chapter 5 - Alternative Exact Model**

  This chapter introduces an alternative mathematical model for allocating thesis defences in the department, where the composition of the committees for each defence is changed. In this model, the committee formation is carried out simultaneously with the allocation process. The chapter elaborates on the indices, parameters, variables, constraints, and objective function necessary to define the model.

- **Chapter 6 - Results**

  Chapter 6 analyzes the results obtained through the two models, with the instances generated. The results are analyzed individually, and a comparison is elaborated between the two. Conclusions about their effectiveness and efficiency are conducted.

- **Chapter 7 - Conclusions and Future Work**

  Chapter 7 brings together all the information collected throughout the study, presenting the conclusions of the same and the guidelines for possible future work.

2. Problem Contextualization

Chapter 2 provides a contextualization of the problem, presenting a broader view of the environment where it is situated and specific details about the case study. Section 2.1 describes the university, IST, and the regulations governing thesis defences. Section 2.2 introduces the department, DEG, and the course related to the thesis defences, MEGI. Lastly, section 2.3 summarizes the conclusions drawn from the chapter 2.

2.1 Instituto Superior Técnico - Master thesis defences

The University of Lisbon is the biggest university in Portugal constituted by 18 faculties and institutes, that offers 216 master's degrees, of which 7 are integrated, to the student community. One of the institutes that constitute this university is IST, a school founded in 1911, with more than 11000 enrolled students, in the 2021/2022 academic year [1]. IST has three campi, being them Tecnológico e Nuclear, Taguspark and Alameda. The oldest one is Alameda Campus, and is in there that the majority of the students is enrolled, being 9808 students (2021/2022 academic year). Taguspark Campus is a more recent building, situated at a distance of almost 20 km from Alameda Campus, with 1526 students. There is a wide variety of master's degrees available, taught at this faculty, 30 of which take place at Alameda, 5 take place at Taguspark, 1 is taught online, 2 is thaugh at Alameda Campus and Tecnológico e Nuclear Campus and 1 take place at Alameda Campus and Taguspark Campus. The number of students enrolled at this faculty has increased in these two last years, as has the number of graduates at the end of each year. The latest numbers, provided by the university statistics and prospective unit, about the graduate students refer to the 2020/2021 academic year, which show that 1305 students became masters in that year, 1131 enrolled in Alameda and 174 in Taguspark [2]. Due to the decree-law 65/2018, published on 16 October [3] , a restructuring in the teaching practices and in the structure of the courses was implemented, with only one master's degree remaining integrated, the master's in architecture. All other masters, which were previously integrated, are now separated from the undergraduate programs.

This decree-law reference that to have a master's degree, a dissertation, specialized in a scientific subject

[1] *About Técnico at Instituto Superior Técnico* (2023). Accessed: 2023/04/28. https://tecnico.ulisboa.pt/en/about-tecnico/
[2] *Balanço RAIDES 2021-22 (2022)*
[3] *Decreto-Lei n.º 65/2018 de 16 de Agosto do Ministério da Ciência, Tecnologia e Ensino Superior (2018)*, Diário da República n.º 157/2018, Série I.

has to be elaborated and presented to a committee. The dissertation is equivalent to 30 credits, and the student has to be oriented by doctors or specialists in the field of it. Furthermore, the discussion of the research done has to be presented to a committee, constituted by three to five members, or five to seven members, if the dissertation is done in association with an international faculty. The committee is chosen by the institution of higher education where the student is enrolled, and one or two members of it can be the student's mentors. The members of the jury decide, through a reasoned vote, in which all members have the duty to vote, therefore abstentions are not allowed, on the grade that should be attributed to the thesis presented.

In IST there are some rules defined for all the courses, regarding this matter [4]. The dissertation can be carried out in different areas, such as a scientific thesis, an internship in a company, or a multidisciplinary Capstone Project. The project should study new problems and methodologies, propose solutions for the problem presented, demonstrate the results of the implementation of the solution and critically analyze them. The available themes, for the dissertations, are submitted, on the official platform of the university, Fénix, by the invited professors of each IST department. These themes are approved by the scientific committee of the different courses, and as mentioned in the decree-law, the orientation of the thesis is done by a professor or researcher of the university. As an exception, when the student realizes the dissertation in a business environment, a confidentiality and an intellectual property agreement has to be signed, as well as an internship agreement. Regarding the jury, it must be made up of three or five elements, a chairperson, a supervisor, and an additional member. The chairperson must be the course coordinator or a member of the scientific committee and is also important to mention that the student's supervisor can not be the chairperson. The committee will deliberate about the grade to be attributed to the student, whereas, in the event of a tie, the chairperson has the quality vote. The thesis defence cannot exceed the limit time of 90 minutes, being the recommended time of 60 minutes for the defence. The time must be well managed, by the chairperson, with 20 minutes being allocated to the student's presentation, and the remaining time must be to discuss the study done, with the intervention of the jury and the student.

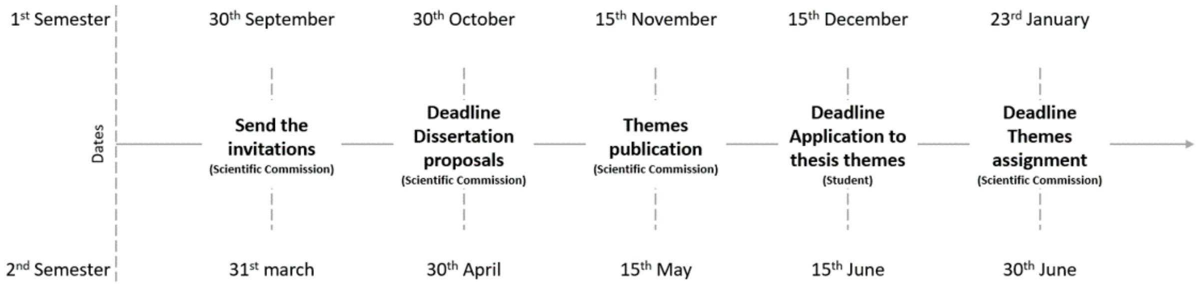
The thesis theme, as mentioned before, can be offered by the different invited professors, or the student can purpose a scientific theme, that has to be approved by a professor in the scientific committee [5]. The supervisor should be the professor who approved the topic. Firstly, the invitation for the submission of topics is sent to the professors, by the scientific commission or the course coordination. Then the themes are published in Fénix, also by the scientific commission or the course coordination. The next phase, is duty of the student, who is responsible for applying to the topics, in the same platform. The last step is the selection of the students and the themes attribution. In a academic year, there are two phases for the students apply to the different theses themes, one in the first semester and one in the second semester. Usually the students that apply in the first semester, do the thesis defence in the second semester of the same academic year. All the steps, mentioned before for the submission, application and assignment of

[4] *Regulamento das Dissertações de Mestrado do Instituto Superior Técnico da Universidade de Lisboa (2022). Accessed: 2023/04/28*

[5] *Temas da Dissertações de Mestrado - Prazos e Tramitação* (2022). Accessed: 2023/04/24

thesis themes have pre-defined dates, that can be consulted in Figure 2.1 which usually remain constant from one academic year to the next.

Figure 2.1: First and second semester dates of the steps for the submission, application and assignment of Thesis Themes



Source: *Temas da Dissertações de Mestrado - Prazos e Tramitação* (2023)

Deadlines for the steps related to the dissertation conclusion are also defined by the institution[6] for all the masters of each department and can be consulted in Figure 2.2. There are also two phases for the dates of the steps related to the conclusion of the dissertation. The process starts with the student delivering the project in the platform Fénix, and the scientific commission or the course coordination should approve the jury, for each thesis defences, in a range of 2 weeks. After that, each department has 1 month to do all the discussions of the dissertations delivered, attributing a grade to each one. The next step is the upload of the final version, being the student's responsibility. After the confirmation of all the documents, by the scientific committee or the course coordination, the grades are published.

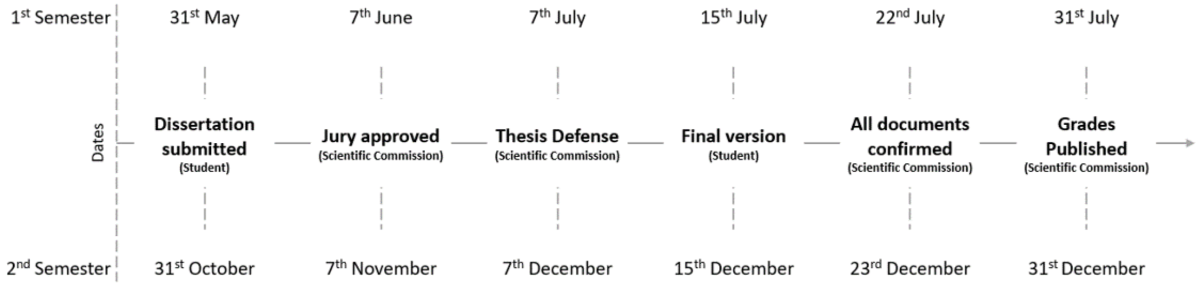
2.2 Industrial Engineering and Management - Master thesis defences

The MEGI is offered by DEG, that also offers a Master´s in Engineering and Management of Innovation and Entrepreneurship. The DEG, most recent department of the IST, founded in 2001, is organized in two scientific-disciplinary areas: Systems Engineering and Management and Engineering and Management of Organizations. This department focuses on the needs of the companies and the markets, offering undergraduate, postgraduate, master's and doctorate programs, that will graduate workers capable of apply different strategies, plans and projects, in the most varied set of markets [7].

This dissertation focus its study in the DEG, more particularly in the timetabling processes of the MEGI

[6] *Conclusão da Dissertação de Mestrado - Prazos e Tramitação* (2022). Accessed: 2023/04/24
[7] *Presentation at Department of Engineering and Management* (2023). Accessed: 2023/04/28. https://deg.tecnico.ulisboa.pt/en/about/presentation

Figure 2.2: First and second semester dates of the steps for the Thesis conclusion



Source: *Conclusão da Dissertação de Mestrado - Prazos e Tramitação* (2023)

thesis defences. This department is located in the Alameda Campus and in the Taguspark Campus, however the master mentioned has almost all their classes, evaluations, thesis defences and presentations in Taguspark Campus. To be accepted in this master, the student should follow some requirements, defined by the university, such as hold a first degree in Science and Technology or hold an academic, scientific or professional curriculum, that proves that the student has the necessary background and skills to carry out the master's degree.

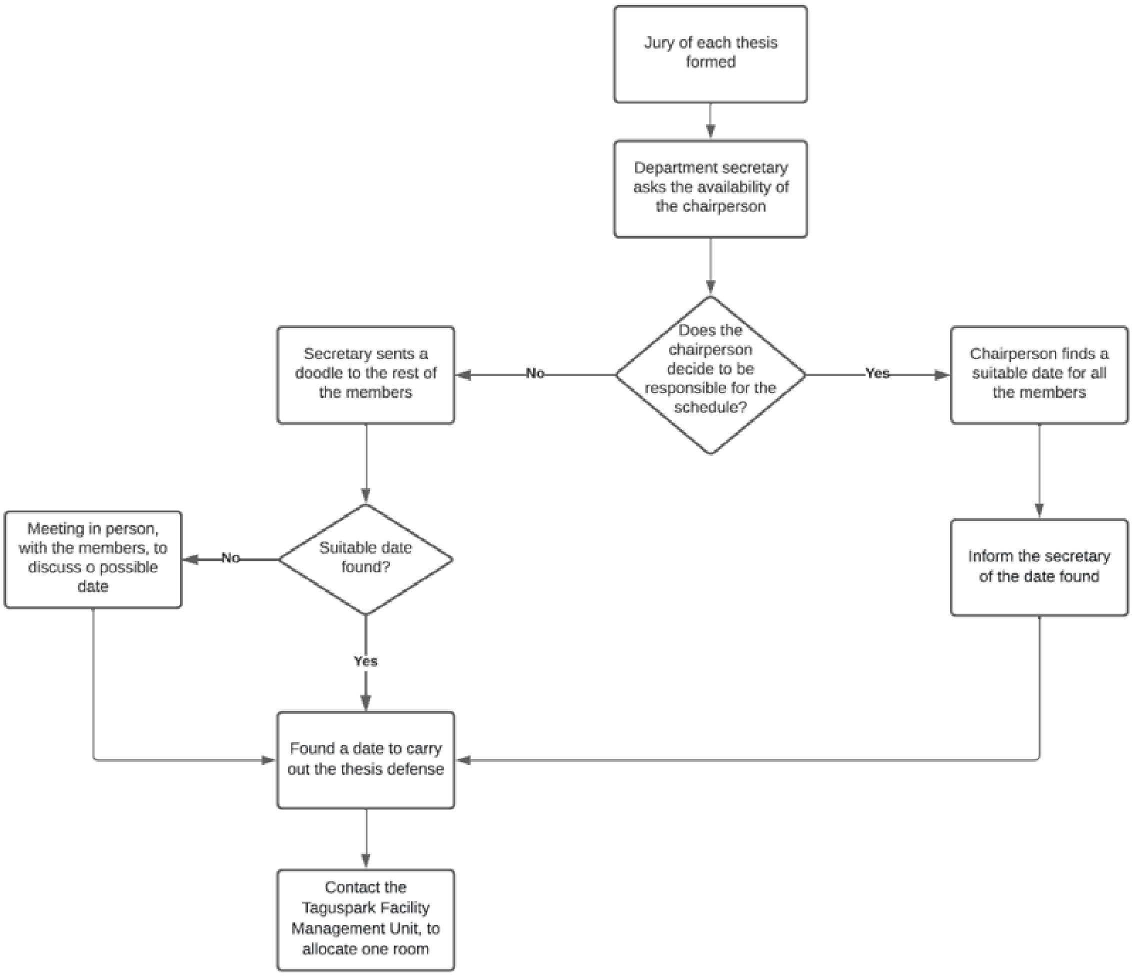
As indicated before, there was a restructuring in IST master's, due to the decree-law, 65/2018, which made the master's study plan change [8]. Currently, the master's degree core structure is divided into two areas, possible of expertise: Operations and Logistics and Financial Management, which corresponds to 72 credits. In addition, to complete the master is mandatory finalize 18 credits of free options, courses that the student's can choose, regardless their area, and conclude a scientific dissertation, corresponding to 30 credits. Following the normal study plan, structured by the department, the dissertation can be completed in the first or in the second semester of the second year, and is recommended that he student has done at least 60 credits. Before the restructuring implemented in the department, 12 credits were attributed to a course denominated by Project in Industrial Engineering and Management. In this course, the student should prepare and present a project, where he reflects about the development of his thesis. The student has to present his work, to a jury with the same constitution of the jury that evaluates the thesis defence. Therefore, until the academic year of 2020/2021, the MEGI coordination was in charge of forming the juries for each student's project, that were the same that for the dissertation. From year 2021/2022 onwards, without the course Project in Industrial Engineering and Management, the department only makes the constitution of the jury when the dissertation is delivery, in the official platform Fénix.

At the moment, the timetabling of the thesis defences, of the MEGI, is carry out by the department's secretary. After the jury of each thesis, delivery in the system, is formed, the secretary starts the process

[8] *Programme Overview at Instituto Superior Técnico* (2022). Accessed: 2023/04/28. https://tecnico.ulisboa.pt/en/education/courses/masters-programmes/industrial-engineering-and-management/

to schedule each defence. As said before, the formation of the committee follows the same rules as those established for all IST departments, consisting of 3 to 5 members. Besides the chairperson and the supervisor, other member is present in the committee, and most of the time it is the supervisor who suggests this other member to the coordination of the department, so that he/she makes part of the constitution of the jury. At first, the availability of the chairperson is collected, by the secretary. Then, a doodle is formed, and sent to the rest of the members, in order to try gather common availabilities between the three members. Once a date is found, suitable for all of them, it is necessary contact the Taguspark Facility Management Unit, to allocate one room to the do the thesis defence. All these steps can be consulted in Figure 2.3. Since the defences have a deadline to be executed, there is a high pressure in he department secretary, to execute the scheduling. There are some exceptions where the chairperson decides to be responsible for the schedule, being in charge for talking to the other members, finding a common date, and communicating it to the secretary. Another exception is when is not possible find a common data, for all the members of the jury, through the doodles and emails sent. In these cases, the secretary meets in person with the members of the jury in order to find a date for the defence of the thesis. These cases take some effort and necessary time from the secretary to spend.

Figure 2.3: Steps for scheduling the master thesis defences



With the process being designed in this way, problems can be identified, that remove some efficiency to the process, making it slower and a greater need for effort on the part of the person responsible for this scheduling. The identified problems can be enumerate:

- Once the process takes some time since the chairperson gives his/her availability until a date is defined, sometimes the availability of the members changes throughout the schedule, which will interfere in the dates defined in the doodle, sent to the other members.
- There may be times that the members of one committee belong to committees from different departments, that will try to schedule the defences in the same space of time, which will make the process even more difficult.
- When is not possible found a common date for the three members, through the doodle sent, it is necessary for the secretary to exchange a large amount of emails with the members, increasing the effort she has to allocate to the scheduling process.
- The dates found with the the doodle do not take into account the distance between the two campi, Taguspark and Alameda, neither take into account the optimization of teachers' time, that is, do not try to minimize idle time, nor the displacements that teachers have to make to the campus Taguspark.
- With the defences back to the normal way, being in person, the allocation of a room is mandatory Sometimes, when the secretary informs he Taguspark Facility Management Unit, there are no more rooms available to carry out the defence, and another date has to be found.

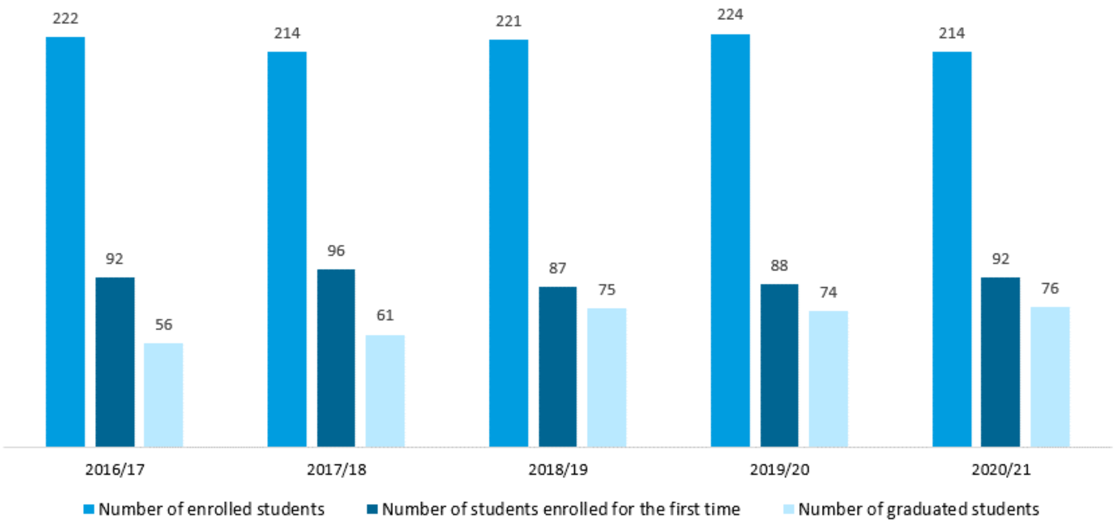
According to the statistical data[9], which can be consulted in Figure 2.4 from the academic year of 2016/2017 to 2020/2021 a average of 219 students were enrolled in MEGI each year, with 91 students enrolled for the first time. The average number of graduates, between the academic year 2016/2017 and 2020/2021 is 68.

2.3 Conclusions

The allocation of thesis defences is an essential process carried out in all IST departments responsible for master's degrees, subject to the same regulations and deadlines from the selection of topics to the submission of dissertations by students. As expected, the DEG must also allocate thesis defences twice a year, a task currently performed manually by a single individual. Given the considerable number of students submitting their dissertations each semester, the collection of availability data from committee members, and the restricted time window available for allocation, the problem's complexity is significantly high, placing a heavy overload on the individual tasked with it. In addition, the manual allocation process suffers from several issues, further impeding its efficiency. As a result, this dissertation aims to optimize the thesis defence allocation process in DEG, initiating the research in timetabling problems, particularly

[9] *Balanço RAIDES 2021-22 (2022). Accessed: 2023/04/24*

Figure 2.4: Number of students enrolled and graduated in MEGI



in thesis defence timtabling problems.

3. Literature Review

Chapter 3 provides a comprehensive review of existing Timetabling Problems and their broad applicability. In section 3.1, this type of problem is contextualized, giving an overview of its nature. The subsequent section, 3.1.1, conducts a state-of-the-art analysis of ETP, defining the problem, presenting the typical constraints, objective functions, approaches, and methodologies developed over the years to tackle this issue, and highlighting its applications. This type of problem encompasses School Timetabling Problems (STP) and University Timetabling Problems (UTP), which are presented respectively in sections 3.1.1.1 and 3.1.1.2. The allocation of thesis defences problems at universities is a part of UTP, together with the University Course Timetabling Problems (UCTP) and the University Examination Timetabling Problems (UETP). All these problems have been analyzed. Finally, section 3.2 presents a conclusion of the conclusions obtained in the analysis of the state of the art carried out throughout the chapter.

3.1 Timetabling Problems

Timetabling problems can be defined as the allocation of a given resource to a timeslot, so that this allocation respects as much as possible the constraints defined, and the established objective has to be satisfied (Wren, 1995 [144]). These problems are made up of finite sets of times, resources, events, and constraints. The sets of constraints to which the problems are subject have evolved significantly, due to technological advances, which have made the problems more specific and hard to formulate. Therefore, it becomes increasingly difficult to solve this problem manually, so that the solution respects all constraints and is satisfactory for those who will make use of it.

This type of problem has been studied for several decades, with the beginning of this investigation given by Gotlieb, in 1963 [58], followed by several articles on the same topic, with an increase in the quality and diversity of the models presented, as well as in the greater focus of research on automatic scheduling solutions (Schaerf, 1999 [119]).

The timetabling problem is applied to the most varied situations and industries, taking as an example the Train Timetabling Problem (Cacchiani & Paolo, 2012 [28]), the Sports Timetabling Problem (Trick, 2000 [137]), the Employee Timetabling Problem (Meisels & Schaerf, 2003 [83]), and the Educational Timetabling Problem (Pillay, 2016 [97]).

The type of solutions found for these problems may differ, depending on the way the problem was formulated. In the search problems, the solution found must respect all the constraints imposed in its definition. In turn, in optimization problems, the solution is found, satisfying all hard constraints and respecting as much as possible the soft constraints, not being necessary condition respect all of them. The quality of the solution found is measured based on the number of respected soft constraints. The higher this number, the better the solution is. For each soft constraint not satisfied, an added penalty (Tan et al., 2021 [132]) so that the solution found can be evaluated.

These types of problems are classified as NP-hard problems (Babaei et al., 2015 [17]), which illustrates the complexity associated with this type of problems. Some of them can be considered NP-complete, depending on the type of constraints defined (Pillay, 2014 [98]). This group categorizes the most challenging problems to solve efficiently.

3.1.1 Educational Timetabling Problems

Within the types of timetabling problems mentioned above, the study of ETP is relevant, which is one of the most studied problems in the set of scheduling problems. In the study to be carried out, it will be considered that these problems can be divided into STP and UTP, as well as in the survey carried out by Babaei et al., 2014 [17]. The first refers to the necessary schedules to carry out in primary and secondary schools, while the second type of problem, as the name implies, refers to the timetables necessary for the proper functioning of universities.

Kristiansen and Stidsen, in 2013, carried out systematic research on the ETP, summarizing in a document the research that had been carried out previously on this topic. The researchers refer to the different types of problems from the ETP, presenting a description of them and the models used to find a solution to the problems. The solutions were tested on real data, and in certain cases [76], turned out to be implemented. In this research, 4 groups of stakeholders related to the educational problem were identified: administration, students, teachers, and departments. All these groups define constraints to the problem, which must be taken into account later. It should be noted that this study refers to a different type of educational problem, the Student Section, which assigns students to different classes and shifts, however since this is not a problem at the university under study, it will not be considered.

In 2016, Pillay [97] carried out a review on the Hyper-heuristics used for ETP, reaching the conclusion that Hyper-heuristics are an efficient way to solve the problem, being able to produce a generalized solution. It should be noted that differences were found in the success of the heuristics used, for each type of more specific problem, i.e the most efficient heuristic for one problem may not be the most efficient for other type of ETP. In this study, the different Hyper-heuristics were divided into 3 degrees of generalization, depending on the type of generalization that the hyper-heuristic allowed to do.

3.1.1.1 School Timetabling Problems

The STP encompass the timetabling problems of schools where several subjects have to be taught, with different teachers to do it, who have to be allocated to different classes of students. This term refers to the construction of weekly timetables for each class, in schools (Schaerf, 1999 [119]), where a certain subject is allocated to a time interval, without overlap of teachers, classes or classrooms. Within the ETP, STP are the least studied by researchers, showing a difference in their development (Pillay, 2014 [98]).

Problem Definition

In this type of problem, it is necessary to consider three types of resources, classes, teachers, and classrooms, which must be allocated to different timeslots, respecting certain constraints (Beligiannis et al., 2008 [21]). Classes are made up of an aggregate of students, formed before the construction of timetables, so it is not necessary to take this into account when formulating the problem. Most of the research carried out considers trios consisting of a teacher, a class, and a classroom, allocating them to the available timeslots, on each day of the week (Birbas et al., 2009 [23]). Sometimes there is not a specific room, assigned before, so the model is also in charge of defining in which room the class will be taught. In these cases, it is necessary to have information about the number of students in each class, as well as the capacity of each classroom (Post et al., 2012 [101]). As in the case of the classroom, the teacher who teaches the subject may not be defined, which is the case when there is more than one teacher capable of teaching the same subject. Thus, more constraints will have to be taken into account in the formulation of the problem.

Problem Constraints

There are several constraints that this type of problem presents, that can be characterized as soft and hard constraints (Carter & Tovey, 1992 [31], Hinchliffe, 1973 [59]), as mentioned earlier. Post et al., in 2012 [101], characterized the different constraints in three groups, namely, basic scheduling constraints, event constraints, and resource constraints. The first group refers to allocating a certain timeslot and resources to an event. The second group mentioned encompasses the constraints that each event, in this case that each class, implies. Finally, the last group refers to resource constraints used in the schedule. On the other hand, Pillay, 2014 [98], divided the constraints of this type of problem into seven categories, namely, problem requirements constraints, non-conflict constraints, resource utilization constraints, workload constraints, constraints period distribution, class constraints, and preference constraints. In total, he mentioned 28 constraints.

Due to the high number of constraints mentioned and taken into account in the existing literature, this literature review will consider the most common constraints in this type of problem, all of them mentioned by Tan et al., 2021 [132]. This article was taken into account, as it was the most recent survey found on the subject.

- R1: Assign a time interval to the selected event;

- R2: Assign a resource to the selected event;
- R3: Schedule the selected resources without conflicting them, that is, each resource can only be assigned to one event in the same timeslot;
- R4: Limit the number of hours each resource is used per day, in the case of teachers and classes;
- R5: In case there are several teachers capable of teaching the same subject and this subject is taught individually, the assigned teacher must always be the same for each class;
- R6: When there is a preference in the time interval for a given event, it must be assigned to it;
- R7: When a resource type is preferred for a given event, it must be assigned to it;
- R8: Consider a time limit that selected resources can be unused;
- R9: Minimize the number of times the same event, from a given class, is assigned on consecutive days.
- R10: Scheduling must be done in such a way that each subject is not taught more than once a day, for each class.
- R11: In case there are sub-events taught on the same day, the time interval assigned to each one must be consecutive.
- R12: Assign the time intervals chronologically, so that there is a better organization of the schedule.

Objective Functions

Throughout the existing literature on timetabling problems in schools, it is possible to highlight two types of objective functions. Both functions, present in the different studies, take into account the hard and soft constraints defined for the problem under study, however, they differ in the way in which the cost calculation of each solution is performed. Several researchers formulate the objective function as the sum of constraints not respected by the solution found, trying to minimize this number (Valouxis & Housos, 2003 [139], Wood & Whitaker, 1998 [143]). On the other hand, different researchers define the objective function in such a way that each constraint violation has a different weight associated, thus assigning different importance to each constraint (Kristiansen et al., 2015 [75]). In this case, the solution found will give priority to constraints that have a higher weight compared to the others (Nurmi & Kyngas, 2008 [91], Colorni et al., 1998 [38]). The objective function can also be defined as a single or a multi objective function, depending on the number of objectives considered in the function, as the name says, however, all the studies that were taken into account defined single objective functions for the STP, with the exception of the study produced by Nurmi & Kyngas, 2007 [92], that presented a multi objective function.

Approaches and Methodologies

There are several approaches to solving this type of problem. There are two groups to highlight, in the classification of existing methodologies, the exact and non-exact methods. Exact methods are those that find the optimal solution for a given problem (Jourdan et al., 2009)[68]. Examples are the Integer Programming Models, Multicommodity Flow Models, Constraint Programming Models, and the Mixed Integer Programming Models. These methods should only be applied to problems with a small number of cases, and since they are time consuming, they should not be applied to NP-hard problems.

As an example of exact methods, we have Branch and Bound applied in 1969 by Lawrie [77], having been the first researcher to find a feasible solution to the problem through the mentioned method. In 2008, Wilke & Ostler [141] applied the same method to a real problem of scheduling in schools. The computational time of this method is quite extensive, so the researchers limited its use to eight computational hours. At the end of this time, the best solution found added up to 3300 penalties due to non-respected constraints. Furthermore, the minimum constraints, once they were considered soft constraints, were violated 128 times, which makes the generated solution not valid. Birbas et al., 1997 [24], formulated the problem as an Integer Programming model, which allowed them to model the different constraints necessary for the solution found to be feasible and optimal. The model built is flexible, which means that it can be adapted to different schools, in addition to the one in which it was tested, to incorporate the different requirements of each institution. Later, in 2009, Birbas et al., 2009, [23] resumed the study of this type of problem, again using Integer Programming approaches, having obtained excellent results, such as compact schedules, without conflicts and with well distributed classes, which is one of the quality criteria. Boland et al.,2008 [25], also used Integer Programming Models to formulate his problem, however, unlike other researchers, he considered constraints in his model so that it was able to form groups of students, depending on existing limitations. In 2003, Papoutsis et al. [95] used the Column Generation approach in the scheduling problem in schools, since it had already been quite effective when tested in airline crew scheduling problems. This method has demonstrated high levels of effectiveness. To reduce the computational time that exact methods usually need to find the optimal solution, Ribic & Konjicija, in 2010 [110] formulated a two-phase model using Integer Programming. The first phase of the model consisted of choosing the day of the week, and the second phase of the model determined the schedule that should be made for that day. Sorensen & Dahms, in 2014 [125], also applied a two-phase model to the same type of problems, consisting of the assignment of a given class to a time interval and a second phase to a class. This solution obtained a much higher degree of efficiency compared to the results verified with the original Integer Programming Model with three binary variables.

A method based on Mixed Integer Linear Programming was used by Santos et al., in 2012 [116], applying a Cut and Column Generation algorithm, using Fenchel cuts. The research developed for the first time revealed strong lower bounds for a type of scheduling problem in schools, called the Class-Teacher Timetabling Problem with compactness requirements. In 2015, Kristiansen et al. [75] uses the same method. The formulated model allowed the researchers to find, for two different cases, solutions that had not been possible to calculate before. In 2015, Al-Yakoob & Sherali [145] proposed a two-phase decomposition of a Mixed Integer Programming Model, considering the model unsustainable when applied to a realistic number of instances. The model, in the first phase, allocates a time interval, of the week, to each class, and in the second phase, it determines the teacher who will be responsible for teaching the subject to the selected class. The researchers also proposed that the selection among the existing feasible solutions be carried out through a Mixed Integer Programming formulation. Due to the high number of variables present, the Column Generation method was applied. The last mentioned method proved to be more efficient, using fewer teachers to schedule in schools, in addition to being less time consuming compared to the two-phase approach. Fonseca et al., in 2017 [56] presented new formulations and cuts off the existing Integer Programming Model. There was a great improvement in the linear relaxation of the formulation, drastically increasing its lower bound, through the added cuts. It was also concluded that this was the most effective formulation for the majority of cases where it was applied. Dorneles et al., in 2017 [47], proposed a multi-commodity flow model, applying a Danzig-Wolf decomposition, also proposing a column generation algorithm, to reduce the time the model takes to provide strong lower bounds. The researchers decided to focus their study on a sub-problem of the STP, the Class-Teacher Timetabling Problem with Compactness Requirements. The lower limits found were the best for 5 of the 12 instances tested, compared to the pre-existing ones, presenting the advantage of being a simpler and faster method. In 2020, Tassopoulos et al. [133] proposed a model of Mixed Integer Programming, to be applied in 10 instances in secondary schools in Greece. Two types of solvers were used, Gurobi and CPLEX. Two methodologies were proposed, the first one taking into account all existing soft and hard constraints, while the second divided the problem into six sub-problems. The study concluded that the first methodology was not successful, even with high computational times. However, with the second methodology applied, the results obtained were quite satisfactory.

As the timetabling problems in schools are constituted by several constraints, it is also possible to apply Constraint Programming. Marte, 2002[81], Valouxis & Housos, 2003 [139] and Demirovic & Stuckey, 2018 [43] developed studies that applied this approach to this type of problem.

Sometimes, when the number of cases is very high, the best option is to opt for non-exact methods, such as heuristics. Starting with the analysis of meta-heuristics, there is a lot of research done in this field, applied to the School Timetabling Problem. This method can be divided into a population based-algorithm and a non-population-based algorithm.

In the population-based algorithm category, within the meta-heuristics, there are evolutionary algorithms. Colorni et al., 1990 [37], applied a genetic algorithm to a scheduling problem in an Italian school. The researchers defined genetic operators in such a way that the cost functions, which assign penalties to infeasible solutions, were minimized. Likewise, Abramson & Abela, in 1991 [4], proposed the same type of algorithm, in which the objective function is a cost function that counts all the times that the formed schedule overlaps resources. In 1999, Fernandes et al. [52] also applied a genetic algorithm. The problem was formulated as a chromosomal representation, using a repair function, so that the solution was valid. Hard constraints were considered, such as the impossibility of assigning a teacher, a class, and a classroom to the same time interval (constraint R3 mentioned above), the non-assignment of a resource to a time interval, previously defined as not available, according to previously established preferences and lunch breaks must be respected. Filho & Lorena, 2001 [53], applied a variation of the genetic algorithm, the constructive genetic algorithm, in two Brazilian schools. Nurmi & Kyngas, 2007 [92], like the aforementioned researchers, again applied a genetic algorithm controlled by nine different parameters. Raghavjee & Pillay, 2008 [105], propose a genetic algorithm, with the objective of applying it to different cases, instead of focusing only on one school, in order to compare the results obtained with those of other studies. In 2012, Domros & Homberger [46] developed an evolutionary algorithm, used in the third international scheduling competition, where the number of constraints is substantially greater than the one for which the other genetic algorithms were tested. Two ideas were implemented in the solver, an indirect representation of the times, through a permutation of the sub-events, and an evolutionary research controlled by an evolution strategy. Recently, in 2016, Sutar & Bichkar [131] proposed a simple genetic algorithm with augmented knowledge operators and probabilistic repair in the data crossing. Once again, the objective function is to obtain schedules that have a total of zero overlaps of resources and must respect the number of times that each set of teacher, class, and classroom must be allocated. The Genetic Algorithm operators were developed with dominant knowledge. They concluded that the simple genetic algorithm found feasible solutions, however, the genetic algorithm obtained faster solutions, under the same conditions.

It is also important to analyze meta-heuristics classified as non-population-based algorithm, such as the adaptive large neighborhood, variable neighborhood search, parallel local search, tabu search and simulated annealing methods.

In 2012, Sorensen et al. [126] proposed an algorithm based on Adaptive Large Neighborhood Search. The algorithm made it to the second round finalist of the international calendar competition. Three strategies were applied in the algorithm: remove strategy, adaptive strategy, and accept strategy. In the same year, Sorensen & Stidsen [127] applied the same type of algorithm in secondary schools in Denmark. This algorithm included a different constraint from the usual ones, the requirement that each teacher has a certain number of days off work. In implementing the algorithm, two neighborhood operators were used: insertion, used to insert events, and removal, used to remove or reallocate resources and events according to the pre-established objective.

Saviniec et al, 2013 [117], developed two different algorithms, using the variable neighborhood search method. It included two neighborhood operators, to solve the problem heuristically: torque and matching operators. Both algorithms proved to be efficient in solving the problem, and the solutions found are very close to the optimal solutions. In 2014, Fonseca & Santos [54] proposed different algorithms based on variable neighborhood search. They applied different variants: reduced variable neighborhood search, skewed variable neighborhood search, and sequential variable neighborhood search. It was concluded that the skewed variable neighborhood search was the most efficient, compared to the other proposed algorithms.

Saviniec et al., 2018 [118], presented two methods in which he applied parallel local search. The solution chosen is the one that minimizes the sum of penalties incurred due to violation of soft constraints. Two strategies were used: central memory-based (solution is produced by a set of meta-heuristic agents, and this is kept in central-memory) and diversification and intensification of memory-based.

Tabu search is another meta-heuristic that researchers put into practice in order to find the best scheduling solution in schools, and most of the time they present better results than the other meta-heuristics (Santos et al., 2004 [115]). Alvarez et al., 2002 [11] used tabu search to determine teachers' schedules in a Spanish secondary school. To determine the necessary parameters to be used in the algorithm, multiple regression techniques were used. The problem is solved in two parts, first, they are allocated to blocks of subjects, allocating teachers to simple subjects in a second phase. The algorithm proved to be flexible and a strong tool for obtaining high-quality solutions. Likewise, Santos et al., 2004 [115] presented two memory-based diversification strategies. In order to prevent the search from getting stuck at certain points in the search space, two approaches were put into practice: adaptive relaxation (changes the values of the objective function) and random restart (the information obtained previously is no longer used, generating a new solution). In 2006, Jacobsen et al. [64], proposed a tabu search algorithm, applied in a German secondary school. The algorithm was tested in 1500 cases. In this algorithm, two types of neighborhoods are applied alternately, period-neighborhood and room-neighborhood. Minh et al., 2010 [87], applied a tabu search algorithm to achieve an optimization of the schedule previously obtained through a greedy search.

Simulated annealing is a meta-heuristic, as mentioned earlier. Abramson, 1991 [3], used this Monte-Carlo optimization technique. A parallel algorithm was presented, possible to be implemented in a multiprocessor. It was concluded that this was more efficient than when an equivalent but sequential algorithm was applied. Melicio et al., 2006 [84] used two types of algorithms, complementary to each other, an iterative algorithm based on fast simulated annealing and a heuristic constructive algorithm. This method implemented a quick evaluation of the new solutions found. Liu et al., 2009 [80], also applied a simulated annealing algorithm, with the difference that it has a new neighborhood structure. Likewise, Zhang et al., 2010 [147], applied a new neighborhood structure using an algorithm based on simulated annealing. Instead of making changes in the allocation of events, as is usual in the standard model of simulated annealing, it makes changes at the level of time intervals to be selected.

There are researchers who have applied matheuristic methods to the STP. This is a different type of heuristics, which explores the application of mathematical programming techniques to heuristics and meta-heuristics. In 2014, Dorneles et al. [48] proposed a Mixed Integer Linear Programming Model and a fix-and-optimize heuristic combined with a variable neighborhood descent method. The method aimed to minimize the teachers' working days, as well as the minimization of idle time, present in the timetables. The Mixed Integer Linear Programming Model is used in order to find feasible solutions to the problem since it was formulated taking into account its constraints. In a second phase, the previously found solution is improved through the fix-and-optimize heuristic combined with a variable neighborhood descent method. In 2014, Sorensen & Stidsen [128] also proposed the combination of meta-heuristics and Integer Programming to solve this type of problem. Fonseca et al., 2016 [55], proposed a matheuristic, combining the variable neighborhood search algorithm with mathematical Programming-based neighborhoods. The proposed algorithm used a Kingston High School Timetabling solver to generate an initial solution, applied a variable neighborhood search algorithm to improve the initially proposed solution, and finally applied a matheuristic method to find the best solution in the proposed time. 15 out of 17 solutions were improved through this approach.

Hyper-heuristics are a type of heuristics, sometimes also applied to these types of problems. The goal of Hyper-heuristics is to perform well on most problems, rather than producing the best result for a set of problems (Pillay, 2014) [98]. In 2015, Ahmed et al. [5] evaluated the performance of a series of selection Hyper-heuristics. Selection Hyper-heuristic is used to join nine low-level heuristics, including seven mutational and two hill-climbing heuristics. The study concluded that the efficiency of the hyper-heuristic used is influenced by the different components it consists of, and it was concluded that the random permutation selection method, when combined with an adaptive great deluge move acceptance criterion, has a better performance than the other Hyper-heuristics. Kheiri et al., 2016 [71], proposed a local search algorithm based on a selection hyper-heuristic framework. The proposed approach consists of the improvement of an initial solution, obtained in the first phase through a KHE solver. The improvement carried out in the second phase is carried out through a hyper-heuristic multistage selection, consisting of processes of diversification and intensification. In 2017, Kheiri & Keedwell [70] presented a hyper-heuristic sequence-based selection, characterized by being easy to implement and effective in solving scheduling problems in schools. Again, the KHE solver is used to obtain a first solution, which is then improved through the proposed hyper-heuristic, consisting of a selection component (15 low-level heuristics) and a move acceptance component. It was concluded that, in this type of problem, the selection component is more important than the move acceptance component.

Besides all the methods mentioned before, hybrid approaches are options applied by the researchers to solve these types of problems. Fonseca et al., 2016 [40], presented a hybrid local search to solve STP. The researchers developed different structures for the neighborhood, to be applied in a hybrid meta-heuristic that mixes simulated annealing with iterated local search. These two approaches are used in the model to improve the solution found. First, an initial solution is produced by the Kingston High School Timetabling Engine. Then simulated annealing is applied to improve the solution, and in a third stage, iterated local search is used to make the solution even better. Demirovic & Musliu, 2017 [42], proposed a combination of local search with a novel maxSAT-based large neighborhood search. The initial solution is provided by the local search approach, which works with destroy and insertion operations, being later improved by the large neighborhood search techniques based on maxSAT. This approach proved to be highly effective, outperforming the others results presented in the available literature. In 2017, Skoullis et al. [122] presented a hybrid cat swarm optimization. The model proposed proved to be more efficient than all the others approaches, based on meta-heuristics, tested before on the same data.

Application

Some of the approaches presented by the researchers were applied in real cases, most of them to improve the school where the research was done. The other ones were tested in some fictional instances, such as the International Timetabling Competition (ITC) of 2007 instances or Lectio instances.

The information mentioned before about STP can be consulted, all summarized, in Appendix A, for easy categorization of the analyzed researches, where the types of existing problems, solution approaches presented, their details, and in which data the model was tested, are presented.

3.1.1.2 University Timetabling Problems

UTP are usually divided into two types of problems: University Course Timetabling Problems (UCTP) and University Examination Timetabling Problem (UETP). Due to the theme of the study to be carried out, the Thesis defence Timetabling Problems (TDTP) will also be analyzed. All universities face this type of problem, being forced to spend time and resources to carry out the schedule of it. Like STP, these are also made up of soft and hard constraints, and it is strictly necessary that the solution respects all the hard constraints defined so that the solution is considered feasible. The set of these constraints will vary from university to university. Over the years, there has been an increase in the application of algorithms based on meta-heuristics, due to the complexity of the problems.

- **University Course Timetabling Problems**

The UCTPis similar to the STP, being one of the combinational optimization problems, with hard and soft constraints. Universities face this problem whenever a new semester/quarter starts. Once again, due to the complexity of the problem, this task is very time-consuming.

Problem Definition

In this type of problem, it is necessary to schedule the events (lecturers or courses) and assign the different resources: rooms, professors, and students, to a certain timeslot, without having to overlap them, as in the STP. Due to the exponential growth of this problem, it is classified as an NP-complete problem, which means heuristic approaches are more advised than exact methods [17]. The main difference from the STP is that students from different classes, in the case of the universities, from different degrees, can have access to the same lecture. In this type of problem, the capacity of the rooms is very important, due to the different number of students assigned to each course (Schaerf, 1999 [119]). Some literature mention that this problem can be divided into two approaches: Curriculum-based University Course Timetabling and Enrollment-based University Course Timetabling (Kristiansen & Stidsen, 2013 [76]). The first approach respects the curriculum of each degree, designated by the university. The second one does the weekly timetables taking into account the courses in which each student has enrolled. Certain universities use both types of approaches in different phases of the timetabling task.

Problem Constraints

All the available literature about this type of problem mentions the two kinds of existent constraints: hard and soft. A feasible solution is found when all the hard constraints are respected. The best solution is the one in which the number of soft constraints not respected, is the lower. This type of problem may have some variants, resulting in different constraints for each instance. Schaerf, 1999 [119], highlights the following variants: unavailabilities and preassignments, multiple sections and grouping subproblem (case in which the lectures of a specific course must multiply due to the high number of students), periods of variable length and classroom assignment subproblem. Due to the high number of existing constraints, the most common ones used in the various studies conducted on this topic will be mentioned. Starting with the hard constraints:

- R1: The professor can only be assigned to one event, for the same timeslot;
- R2: The room can only be assigned to one event, for the same timeslot;
- R3: The capacity of the rooms must be taken into consideration when assigned to events;
- R4: Only one event can be allocated in each timeslot, having into consideration the curriculum of each degree;
- R5: When the university has two or more campi, the events followed in the timetable must be taught at the same campus;
- R6: The lunch hours must be respected;

As soft constraints, it is possible to mention the following:

- R7: Professor can show some preference for a specific timeslot, to teach his lecture;
- R8: Professor may request special equipment, that must be present in the room;
- R9: The idle times should be minimized, for the students and the professors.
- R10: Lectures must be well distributed by timetable, avoiding having only one lecture in a day.

It is important to mention that some researchers, to obtain feasible solutions, change some hard constraints to soft constraints, in the formulation of the model. This aspect will depend on the goal of the university.

Objective Function

The objective function for this type of problem is, most of the time, a single objective function that considers the violations of the soft constraints defined in the model. The importance of each constraint is different from university to university. Therefore, in some of the literature already done, each constraint violation has a different weight in the objective function. That way, the best solution will be the one that minimizes that total. On the other hand, for universities that do not show preference related to soft constraints, the objective function will be to minimize the number of violated constraints.

Carrasco & Pato, 2000 [29] did something different related to objective function. To measure the quality of the solution, they considered a multi objective function capable of evaluating the quality of the solution for the students and the professors. The proposed algorithm provides a range of trade-off solutions since the constraints related to the quality of the solution, for the students conflicts with the constraints related to the quality for the professors. Ozdemir & Gasimov, 2004 [93], Ismayilova et al., 2007 [63], Phillips et al, 2015 [96] and Jamili et al, 2018 [65] also included in their models multi objective functions.

Approaches and Methodologies

To find an optimal solution for UCTP, some researchers apply exact methods to this type of problem. Daskalaki et al., 2004 [41] proposed a 0-1 Integer Programming formulation of the UCTP, with the objective function of minimizing the linear cost function. They defined different weights for the violation of soft constraints to define the preferences. Because of this factor, the proposed model showed to be flexible due to the multi-dimensional variables. Avella & Vasil'Ev, 2005 [14], also applied an Integer Programming Model to solve the mentioned problem. A Branch and Bound method was used to formulate the problem. A Set Packing polytope was used, to consolidate the initial formulation, together with a polyhedral structure. This algorithm was able to find the optimal solution for some of the real instances, where it was tested. In 2004, Qualizza & Serafini [104] proposed an Integer Programming approach based on column generation. This method creates subproblems, with associated constraints and preferences, defining weekly timetables for each course. To ensure that the solution found is feasible, the researchers applied a Branch and Price Strategy. Schimmelpfeng & Helber, 2007 [120], proposed an Integer Programming Model with twenty sets of constraints, grouped into four groups: teachers' basic assignment constraints, school-specific requirements, constraints resulting from the institutes' perspective, and constraints related to the teachers' preferences. Bakir & Aksop, 2008[19] formulated the UCTP as a 0-1 Integer Programming Model, with the objective function of minimizing the dissatisfaction of students and professors, with the presented solution. They applied new criteria so that the events allocated in consecutive timeslots were taught in the same classroom. The model presented by Daskalaki et al., 2004, mentioned before, was the guide for this one. In 2012, Burke et al. [26] proposed an Integer linear Programming Model with the application of a Branch and Cut procedure, as a way of reducing the number of used variables. The model presented uses a mixture of binary and general integer variables, and a set of cuts from event/free-period patterns to reach the optimal solution. In this way, the efficiency of the model was improved. Phillips et al., 2015 [96], proposed an Integer Programming Model for solving this type of problem, more precisely the practical classroom assignment problem. Due to the large complexity of this type of problem, some researchers divide the assignment of the resources into two phases: first the allocation of an event to a timeslot, and then the allocation of a room to the same event. This model was also used in some instances of the ITC, of 2007, where the model showed an improvement in the solutions found when compared with the solutions heuristically generated.

In this type of problem, as the previously mentioned STP, the most effective way to find a feasible solution is to use heuristics to formulate the problem. This way, computational time will be much smaller compared with the necessary time that exact methods take to find an optimal solution. Heuristics do not guarantee that the solution found is the optimal one, but they find the one with the best quality, having into consideration the objective function defined.

Meta-heuristics are one of the types of existing heuristics. Starting by analyzing the ones categorized by Population-Based algorithms, the most popular ones, among the research community are genetic algorithms, ant colony algorithms, and memetic algorithms.

The genetic algorithm is one of the most common evolutionary algorithms based on the process of natural selection. In 1992, Colorni et al. [36] investigated the application of a genetic algorithm to solve timetabling problems. The presented algorithm used a hierarchical structure of the objective function to allow easy definition of preferences of the constraints. Genetic operators were used to minimizing the objective function, this way infeasible solutions were penalized. Khonggamnerd & Innet, 2009 [72], to improve the effectiveness of automatic arranging university timetables proposed a genetic algorithm. Three genetic operators were employed, in the formulation of the algorithm: crossover, mutation, and selection. All the solutions presented by this approach were feasible, with a 0,70 crossover rate. Alsmadi et al., 2011[9], to minimize the number of violated soft constraints, maximize the use of rooms and reduce the amount of work that each professor has, by day, proposed a genetic algorithm, with the particularity of this approach uses a machine learning. Compared with other existent work, this approach showed the most comprehensive solution.

In 2004, Rossi-Doria & Paechter [112] proposed a memetic algorithm for the UCTP. This approach used local search in the computation of the solutions. These types of algorithms, instead of mimicking genetic biological evolution, are based on cultural evolution. After a set of possible solutions has been created, through the application of the memetic algorithm, local search is applied to improve them. In a way of generating better individuals, the population is subjected to three stages: selection of better individuals, their reproduction through recombination and mutation, and a replacement strategy. Jat & Yang, 2008[66], also applied a memetic algorithm. This algorithm integrates two local search methods, to solve this type of problem. The proposed algorithm found high-quality solutions for the problem.

Ant colony algorithms are also applied to this type of problem. Socha et al, 2003 [123], presented two different methods to solve the UCTP: Ant Colony System and MAX-MIN Ant System. The two methods are very similar. Each path, constructed by the ants, in each iteration of the algorithm, represents the allocation of the events to different timeslots, based on the amount of pheromone. They only differ in the way they use local search to improve the constructed timetables and how they update the pheromone matrix (the MAX-MIN Ant System has upper and lower limits for the pheromone values). The researchers concluded the MAX-MIN Ant System performs better than Ant Colony System, for all the instances tested. In 2012, Nothegger et al. [90] proposed an algorithm using the method Ant Colony. The proposed algorithm used two matrices to represent the information about the pheromone values. Although the model presented the best solution for some instances, the variation in the quality of the solutions was large, since the algorithm found the worst solution for some instances.

The other type of meta-heuristics, the non-population-based meta-heuristics, also known as a single solution based, are also widely used in the construction of algorithms and methods to solve this type of problem. The most common ones are the tabu search algorithm, the variable neighborhood search, the simulated annealing, and the randomized iterative improvement with a composite neighboring algorithm.

The tabu search, as said before, is one of the meta-heuristics, used as an optimization algorithm. Alvarez et Al., 2002 [10], used this method when they proposed a solution approach to solve the UCTP. The method is formulated with three phases: the first one to construct an initial timetable, the second to improve the timetable built by applying tabu search and local strategies, and the third one to improve the room assignment. In the second phase, to improve the obtained solutions, some moves were added to the model: simple move, swap move, and multiswap move. These moves defined three alternative neighborhoods for the algorithm used. In 2003, Cordeau et al. [39], proposed a tabu search method to solve the problem of timetabling courses in universities. This method is also formulated with different phases. The first one is responsible for finding a feasible solution and the second one has the goal of improving the timetable obtained in the first phase, through a basic tabu search algorithm. The algorithm has into consideration the number of soft constraints violated, trying to minimize it. To improve the algorithm used exchange moves were applied to it. Aladag et al., 2009 [6] studied the effect of four neighborhood structures in the solutions presented by the tabu search algorithm. Simple and swap moves were applied to the neighborhood structure. Two of the four neighborhood structures were new structures proposed by the researchers, combining the two techniques referred to before. Chen et al., 2020 [34], proposed a new tabu search algorithm, with the traditional component of this meta-heuristic, the fact of having a short-term memory structure, but with the difference that this integrates the controlled randomization strategy, in the move acceptance strategy. This strategy was applied to balance the diversification and intensification of the algorithm. The research also included two complementary neighborhoods to help an intensive search and a different evaluation technique to reduce the time the method needs to find a high quality solution. It was showing the results obtained by the algorithm were competitive with the existing ones and, 55 feasible solutions were found between the 60 instances where the algorithm was tested.

Another non-population-based algorithm, from the group of the meta-heuristics, is the variable neighborhood search. Abdullah et al., 2005 [2], used an exponential Monte Carlo acceptance criteria in the approach based on variable neighborhood search. The initial solution is produced by a constructive heuristic, starting with an empty timetable, only taking into account the hard constraints. The obtained solutions were as good as the solutions present in the available literature, mainly for small problems. Nguyen et al., 2011 [89], tested the application of this meta-heuristic, and seven of its variants, on a real-world UCTP. As a way of improving the solution, an accepting criterion was applied. The variant that obtained the best results was the Fleszar-Hindi extension of the basic variable neighborhood search.

Simulated annealing is another meta-heuristic, that performs a stochastic search of the neighborhood search. This heuristic allows choosing a point present in the search space of the solutions at the beginning of the algorithm, even if the objective function presents a worse value. Tuga et al., 2007 [138], used the simulated annealing approach to try to minimize the violations of the soft constraints defined for the problem. Instead of using the basic approach, researchers combined the method with a heuristic based on Kemp Chain neighborhood. Graph-based sequential heuristics were used to create the initial feasible solution in the first phase. In the second stage, this method uses three neighborhood structures: simple, swap, and Kempe chain, which allows a better diversification in the search for a feasible solution. Aycan & Ayav, 2009 [15], proposed an approach based on simulated annealing, and, since the neighborhood structure is crucial in these types of algorithms, researchers compared different types of neighborhood searching named as a simple search, swapping search and simple search-swapping, to see which one was more efficient. The best solution was found with the combination of these three searching algorithms. The run time cost was also computed and compared for each solution found. Ceschia et al., 2012 [33] presented a meta-heuristic approach based on simulated annealing. The local search technique of the approach is done in six different stages, one of which is applying the simulated annealing algorithm with probabilistic acceptance and a geometric cooling scheme.

Randomized iterative improvement with a composite neighboring algorithm is used by some researchers to find solutions for the course timetabling problem. Abdullah et al., 2007 [1], proposed an approach that gives feasible solutions, violating the minimum possible number of soft constraints. The algorithm works on the same base as the algorithms formulated with simulated annealing: if the solution presented is best, it is always accepted, however, if it is not, the solution is accepted with a certain probability. Two types of neighborhood structures were used, in an attempt to make the approach more efficient. The algorithm produced seven solutions that were better than the existing ones. One disadvantage found was the computational time needed to calculate the solutions. This approach is seen to be more effective for small instances.

Besides meta-heuristics, hyper-heuristics is another type of heuristics used to solve UCTP. They can be categorized as selection constructive, selection perturbative, and generation perturbative. The most used one is the selection constructive hyper-heuristics.

As a selection constructive hyper-heuristic, Ross & Marín-Blazquez, 2005[111], proposed a constructive hyper-heuristic for the UCTP. They used a messy genetic algorithm to search for the algorithm, that creates the solution iteratively, which is not common in this type of approach. Three phases were used to do the timetable, where the event, timeslot, and room were chosen in one of the different phases. For each phase, the algorithm provides three different heuristics. The formulated algorithm had a good performance in the data that was applied, showing reasonable solutions. Qu & Burke, 2009 [102], proposed a unified graph-based hyper-heuristic framework, that uses different local search-based algorithms, in charge of searching upon sequences of low-level graph coloring heuristics (largest degree, largest weighted degree, largest color degree, largest enrollment, saturation, and random ordering heuristic). The algorithm schedules the events, one by one, by order of difficulty, to construct the solutions. The hyper-heuristic, formulated with variable neighbor search, found the best results for the benchmark set where was tested. The study concluded that iterative techniques are more effective than tabu search. In 2012, Rossi-Doria & Paechter [113] presented an evolutionary algorithm, with a choice of heuristics used to present a timetable as the solution for the UCTP. Some examples of the heuristics used are the largest degree first, largest color degree first and least saturation degree first. The heuristics used were represented by two different rows, where the first one represents the heuristics used to choose the event that will be allocated. The second row represents the heuristics used to select the room and then the timeslot for the selected event. The produced hyper-heuristic only gave good solutions for two of the five instances, which were tested. They also concluded that when graph coloring heuristics were used, the solutions obtained had low quality, which means that these type o heuristics is not the best to use in this type of problem, mainly when used the data set is not from real cases.

As a selection perturbative hyper-heuristic, Kalender et al., 2012 [69], proposed a selection perturbative hyper-heuristic, using a set of low-level constraint-oriented neighborhood heuristics, as a way of solving the course timetabling problem at Yeditepe University. To test the level of performance of the hyper-heuristic developed, it was tested in six other problems, and the result was compared with the existing solutions in the literature. The framework proposed to use two components: heuristic selection (greedy gradient heuristic) and move acceptance (simulated annealing) methods. The obtained results show that the proposed framework outperforms the other approaches. In 2014, Soria-Alcaraz et al., [129], studied the performance of combining a general modeling methodology with an effective learning hyper-heuristic. The proposed framework used iterated local search hyper-heuristic, and to obtain an autonomous design of algorithms, mechanisms of machine learning were applied. It was formulated with two different phases, that iteratively interact with each other: the perturbation phase and the improvement stage. The first phase works by applying a move operator to the solution and the second phase restarts local search from the perturbed solution, by applying online learning, selecting a low-level heuristic from the pool based on the probability of which heuristic performance. The process of learning the operator, for the second phase, was done with dynamic and static schemes. The results obtained show to be competitive when compared to the state-of-art in the 2007 International Timetabling Competition. In 2016, Soria-Alcaraz et al. [130] proposed an Iterated Local Search approach combined with a hyper-heuristic that generates heuristics based on a fixed number of add and delete operations. The delete operation is in charge of removing events from the timetable and the add operation does the opposite, it adds an event to a certain timeslot, respecting the soft constraints defined. This approach showed competitive results with the reported state-of.art results for the instances in which it was tested.

As a generation perturbative hyper-heuristic, Rattadilok, 2010 [107], studied three modifications in a hyper-heuristic approach. In this study, using a greedy approach, an initial solution is created and improved using a generation perturbative hyper-heuristic. The first modification was in the choice function, it made the learning mechanism, related to the values of parameters in the choice function, automatic. This modification showed improvements in the performance of the choice function. The second modification was related to the range of low-level heuristics used by the framework since ten low-level swap heuristics were used. The third modification aimed to obtain a more efficient framework by constructing a hierarchy of sub-controllers, with the range of low-level heuristics modified in the second phase. The sub-controllers were used to make the configuration decisions. The combination of the second and third modifications turned the framework into a dynamic approach. The hyper-heuristic proposed was tested in the data set from the first international university course timetabling competition.

Another approach used by the researchers to solve UCTP is the use of matheuristics. This method was used for the first time, and applied to this type of problem, in 2018 by Lindahl et al. [79]. The researchers proposed the combination of a Mixed Integer Programming Model and heuristics to solve this type of problem. The Mixed Integer Programming part is responsible for exploring part of the solution space in each iteration. On the other hand, heuristics have the role of creating subproblems and fixing some variables, by using problem specific knowledge, in a try of getting a more efficient model. The results showed that the matheuristic has better getting results than the models using the original Mixed Integer Programming. Mikkelsen & Holm, 2022 [86], described a parallelized matheuristic, which combines methods based on Mixed Integer Programming, in a way of finding lower bounds for the problem. The parallelized matheuristics are used to produce an initial solution and to improve the solution found, through large neighborhood search heuristics, a fix-and-optimize matheuristics. They also proposed a diversification scheme to use when the process stagnates. The advantage of the purposed algorithm is the fact that can be applied to different problems, due to its generality, the presets constraints can be changed, without modifying the structure of the algorithm.

It also importantly mentioned the hybrid approaches applied to these types of problems. These approaches are used to present high-quality solutions. Jat & Yang, 2008 [66] proposed a hybrid genetic algorithm and tabu search approach, that solves the problem of university course timetabling through two phases. The first phase consists in applying a guided search strategy, that directs the search based on the characteristics of the last possible solution found and applying local search techniques to improve the quality of the solution. The second phase uses a tabu search heuristic to improve the solution found in the first phase. Kohshori & Abadeh, 2012 [74], presented three hybrid genetic algorithms to solve UCTP, that combined genetic algorithm, fuzzy logic, and local search algorithms (randomized iterative local search, simulated annealing algorithm, tabu search algorithm). After the models were tested, the genetic algorithm with tabu search proved to be the more efficient one. In 2020, Goh et al. [57] proposed a hybrid local search approach, solved in two phases, one to find a feasible solution and the other one to improve the solution given. To find the solution, in the first phase, a tabu search with sampling and perturbation with the iterated local search was applied. In the second phase, an algorithm based on simulated annealing with reheating was applied. The algorithm proved to be competitive with the other ones existing in the available literature. Rezaeipanah et al., 2021 [109], proposed an improved parallel genetic algorithm with local search. To reduce the number of violated hard constraints and to improve the quality of the solution, a distance to feasibility criterion was used.

Application

As with STP, some of the approaches presented were tested in real cases, most of them to improve the way university course timetables were done. The other ones were tested in some fictional instances, such as the International Timetabling Competition (ITC) of 2007 instances or Lewis instances.

The information mentioned before about UCTP can be consulted, all summarized, in Appendix A, for easy categorization of the analyzed researches, where the types of existing problems, solution approaches presented and their details, and in which data the model was tested, are presented.

- **University Examination Timetabling Problems**

UETP is one type of problem that universities face every semester that takes a huge amount of resources to solve, due to the complexity of the problem. This type of problem also faces hard and soft constraints, that must be respected when trying to find the best solution for the problem. These constraints will vary from university to university, due to the differences in the educational structure of each one.

Problem Definition

The UETP is also considered an optimization problem, as the other problems previously mentioned in this literature review. This problem has the objective of scheduling exams in different timeslots, taking into account their duration, which may vary from exam to exam. Carter & Laporte, 1995 [30], defined UETP as the assigning of examination to a limited number of available timeslots in such a way that there are no conflicts or clashes. Hard and soft constraints are also defined in these problems, and due to the growing complexity of it that makes the problem, that makes the problem be considered a hard problem, exact methods are not recommended, once they will need a lot of computational effort to find the optimal solution. Some researchers consider the UETP quite similar to the UCTP, however, some factors vary in the UETP. In this one, two or more events (exams) can be scheduled in the same timeslot for the same room, the best timetable of exams for the students is the one that has the examinations spread, instead of having all the events compact through the days and each course has one exam per semester, in opposition to the case of UCTP, where which course can have more than lesson per week. This type of problem can be divided into Capacitated Examination Timetabling e Uncapacitated University UETP. The difference between these two types of problems is that the first one takes into account the capacity of the rooms where the exam will take place. Some researchers do the distinction is between the curriculum-based approaches and the enrollment-based approaches. Schaerf, 1999 [119] considered some variants of this problem, having into consideration if the problem is formulated taking into account unavailabilities and preassignments, room assignment, and minimizing the length of the session.

Problem Constraints

To find a feasible solution for UETP, all the hard constraints defined in the model must be respected. Besides that, the number of soft constraints not violated should be the maximum one, to be considered the best solution. The defined constraints will vary from university to university. Two types of conflicts, between the schedule of the exams, were considered, in some studies, the first-order and second-order conflicts. The first-order conflict describes situations when exams, that have students in common, are scheduled for the same time (this is usually considered a hard constraint). The second-order conflict describes situations when the solution for the timetable of the exams has two exams too near to each other, not giving study time for the student (usually considered as a soft constraint). Due to the large number of possible constraints that can be enumerated for this type of problem, only the most common ones will be mentioned in this study:

- R1: All exams have to be allocated to one timeslot;
- R2: All exams have to be assigned to one room;
- R3: Exams with common students enroll cannot be scheduled at the same time;
- R4: Room capacity must be taken into account when assigned to an exam;
- R5: Spread the exams through the timetable;
- R6: If two, or more exams, are scheduled for the same room, the duration of them must be equal;
- R7: Exams with a higher number of students enrolled should be scheduled first;
- R8: Each student should have only one exam each day, however, if it is not possible, the exams should be assigned to the same room;
- R9: The assigned room for the event must have all the necessary resources for the execution of the exam.

Objective Function

The objective function of this problem is, most of the time, the minimization of the number of soft constraints violated by each solution found. In this way, the best solution is the one that respects the higher number of soft constraints. Usually, is attributed a weight to each constraint, to differentiate them in the objective function. This way, the violation of certain constraints will have a higher impact on the objective function in comparison with the violation of other constraints. The weights and preferences may differ from university to university. Although almost all researchers applied single objective approaches, is also possible to find multi objective approaches in the available literature about this theme. With the exception of the study produced by Huede et al., 2006 [60], which used a multi objective function, all the other studies applied a single objective function.

Approaches and Methodologies

To find an optimal solution for UETP, some researchers apply exact methods to this type of problem. In 2006, MirHassani [88] presented an approach that used Integer Programming to develop examination timetables with an improvement in the spread time. The spread time consists of the time that each student has available to study between two exams. Mccollum et al., 2012 [82] presented an Integer Programming Model to solve the problem. This model considers that the examination timetable is done after the students enroll in the courses, in other words, is a post-enrollment problem. Three levels of constraints were considered in the formulation of the problem: the strict-hard (constraints that never can be violated), the relaxable-hard (type of constraint that can only be violated during the process of finding an optimal solution), and the soft (constraint that can be violated, with the addition of a penalty in the objective function). The formulated model was not able to present a feasible solution for the instances where it was tested. In 2017, Cataldo et al. [32] proposed an Integer Programming approach with four sequential stages, for the UETP. The computational time that exact models take to find a solution is usually quite high when the problem is very complex. That way, to try to reduce the complexity, clustering, and patterns generation techniques were used. The four stages are related to the resources and timeslots allocated to each event (exam). The model starts by forming groups of courses, assigning, in the next phase, timeslots and rooms to the groups formed. Then, the rooms and timeslots are assigned individually to each group. In the final stage, the timetable is formed, having into consideration the constraints defined.

Mixed Integer Programming Models are also solutions that the researchers present to find a solution for the UETP. Al-Yakoob ET AL., 2010 [146], proposed a Mixed Integer mathematical modeling approach to find optimal exam timetables. This model takes into consideration the resources that need to be allocated to each exam and the availability of the professors and the students. A CPLEX solver was used to solve the two-stage modeling approach presented. In the first phase, the model finds the best timeslot where the exam must be allocated, and in the second stage, the model allocates professors to monitor the exam.

Constraint Programming is another exact method applied by the researchers, to solve UETP. In 1999, Reis & Oliveira [108] constructed a constraint logic Programming approach to schedule the exams at the university. The proposed model assign timeslots, invigilators, and rooms to the exams. To find a feasible solution for the problem. ECLiPSe, a constraint logic programming system, was used. This system gives access to various constraints solvers that can be used to find the solution. The proposed model is formed by 3 phases, where the examinations are split, the distances between rooms are analyzed and the invigilators are allocated to the exams. Merlot et al., 2002 [85] developed an algorithm that finds a feasible solution through a three-stage method. The Constraint Programming Model is used in the first phase.to found a first solution very quickly, having into consideration the capacity of the rooms and the duration of the exam. Some exams remain unscheduled in this first phase, just being allocated to a timeslot in the next one. Huede et al., 2006 [60] proposed a Constraint Programming Model with a multicriteria optimization. Branch and Bound principles are integrated into the search strategies used by the model, to search more efficiently for the solution. In the model proposed, the quality of the solution is evaluated by analyzing the trade-off between some critical criteria.

Besides the exact methods, non-exact methods are also applied to UETP. Starting by analyzing the population based-algorithm, the evolutionary ones are algorithms used frequently in this type of problem. Wong et al., 2002 [142], proposed a genetic algorithm optimizer that presents timetables without exam conflicts and with a maximum of 2 exams consecutive per student as solutions. More than 85% of the students have 2 or more free periods between the exams. Pillay & Banzhaf, 2010 [100], also presented a genetic algorithm to solve the UETP, with a two-phased approach. The first stage finds feasible solutions, without having into consideration the soft constraints. The second one is in charge of optimizing the solution found in the first phase, through a different genetic algorithm. The presented model tested five low-level heuristics to schedule the exams considering their level of difficulty, instead of scheduling them randomly. In 2013, Innet [62] proposed a genetic algorithm, that employed three genetic operators, crossover, mutation, and selection and a chromosome were designed. After testing the algorithm, the researchers found that the crossover rate has to be 0.75. Jha, 2014 [67], proposed a genetic algorithm that is able to allocate resources, like invigilators and rooms, and timeslots to the course exams. A natural chromosome and genetic operators (crossover, mutation, and selection) were used in the algorithm to find a high quality solution.

Azimi, 2004 [16], presented a comparison between the Ant Colony Algorithm (ACA) with the other three meta-heuristics. The solutions found with the Ant Colony approach were first generated with heuristics and then improved with local search. The researcher concluded that ACA was the algorithm that had the best performance among the three. Eley, 2006 [50] presented two approaches, based on the Ant Colony Algorithm, the MAX-MIN, and the ANTCOL approach. and compared the two of them, adjusting some parameters. The approach that presented the best results was the ANTCOL. in 2014, Thepphakorn et al. [134] proposed variants to the Ant Colony Optimization Algorithm, the Best-Worst Ant System, and the Best-Worst Ant Colony System. To improve the quality of the solutions found, local search was applied, obtaining an improvement of 74,5% retaliated to the performance of the systems used. The Best-Worst Ant Colony System produced the best solution, with the lowest number of soft constraints violated.

Another group of meta-heuristics used in this type of problem is the non-population-based algorithm. Starting by analyzing the studies that applied a tabu search Algorithm to solve UETP. Di Gaspero & Schaerf, 2000 [45] tested different versions of an algorithm based on tabu search, and studied diverse biased selection strategies, to find the best neighborhood moves at each iteration of the algorithm. To obtain high-quality solutions a shifting penalty mechanism, a variable-size tabu list, a dynamic neighborhood selection, and a heuristic initial state were used. These factors have proven to be critical in the construction of the approach. In 2002, Di Gaspero, 2002[44] improved the results achieved by the study mentioned before, applying a novel multi-neighborhood local search algorithm, based on a toke-ring search, combining two kinds of moves. White & Xie, 2000 [140], proposed an algorithm, based on tabu search, with longer-term memory and tabu relaxation technique. Two types of long-term memory were used: recency-based short-term and move-based longer-term memory. An improvement of 34% was observed, in the quality of the solutions. It was found that the length of the tabu list was related to the effectiveness of the model. In this way, quantitative analysis methods were applied to estimate what should be the length of the tabu list. Bajeh & Abolarinwa, 2011 [18] compared Genetic Algorithms with tabu search Algorithms. The researchers concluded that both algorithms had a great performance, obtaining efficiently high-quality solutions, however, tabu search obtained better results.

Another meta-heuristic usually applied to UETP is the variable neighborhood search. Burke et al., 2010 [27], studied the application of variable neighborhood search to the examination problem in universities and concluded that this meta-heuristic has the advantage of easily adapted to the different constraints defined in each problem. Besides that, the researchers applied some variants of the basic meta-heuristic, with diverse initialization methods. This approach uses a huge number of neighborhoods, as a way of connecting the search space and finding a high quality solution. To improve the model, hybridization with a genetic algorithm was implemented. This change in the model, makes the algorithm choose, in a more intelligent way, the neighborhoods that must be used, to find good solutions. Alefragis et al., 2021 [7], proposed an extension of the basic variable neighborhood search to solve this type of problem. This extension uses basic meta-heuristic algorithms that try to improve the solution found before, and pass it to the next one. Different algorithms and neighborhoods were tested in this study, to compare the results obtained. The moves used, i.e the neighborhoods used by the meta-heuristics algorithms, are divided into simple and complex moves, depending on the type of transformation they do in the incumbent solution. The researchers concluded that the choice of active moves and the algorithm parameters influence the efficiency of the model.

Simulated annealing is a meta-heuristic usually applied to UETP. Thompson & Dowsland, 1996 [136], proposed three variants of the basic simulated annealing approach, that solve the study problem in different phases. Duo to the different phases in which the algorithm solves the problem, some decisions previously made, can be undone in the next phases. However, this makes a disconnection in the solution space. To solve this, a Kempe chain neighborhood was implemented, which outperformed the standard simulated annealing. This was only possible because the problem was designed as a graph coloring problem. Thompson & Dowsland, 1998 [135] proposed a two-phased approach based on simulated annealing, to solve UETP. This first phase is the one responsible for finding a feasible solution and is in the second phase that the feasible solution is improved, considering the defined soft constraints. To obtain good solutions, choices about the solution space, the neighborhood, and the parameters that govern the cooling schedule had to be done. After testing different combinations of the choices made about the aspects mentioned, they concluded that the neighborhood used is the most important aspect and the Kempe chain neighborhood is the best one to use, in a way to obtain the most effective model. Leite et al., 2019 [78], proposed a variant of the simulated annealing algorithm, denominated by FastSA. In this approach, the first solution is created using a saturation degree-based heuristic, together with conflict-based statistics. To improve the feasible solution found, an optimization phase was implemented, where modified acceptance criteria, denominated by temperature bins, were used. The researchers noticed that for exams without any accepted moves in the next temperature bin, the probability of having more acceptance movements in the future is very small or zero. In this way, the FastSA uses fewer evaluations than the standard SA, which turns the algorithm more efficient than the other. Bellio et al., 2021 [22], proposed a simulated annealing approach for the UETP, with the combination of multiple neighborhoods. An ablation analysis was performed to identify the most important neighborhoods, between the three basic neighborhoods used and the four new ones. This approach works in two phases, the first phase where a feasible solution is found and the second one where the solution found is improved. The researchers concluded that the new approach was able to find solutions better solutions than the ones presented in the actual literature, in the same amount of time.

Another non-exact method, used in this type of problem is hyper-heuristic. In 2012, Sabar et al. [114]

formulated a graph coloring constructive hyper-heuristic, to solve the UETP in the university. The model is constructed by four low-level heuristics, denominated by largest degree, saturation degree, largest colored degree, and largest enrollment. The exams are scheduled per order of difficulty, in each list formed, i.e the most difficult exam is scheduled first. To do the allocation of the timeslots to each exam in a more efficient way, a roulette wheel selection mechanism was applied. The results obtained with this approach are shown to be competitive with the ones present in the literature. Soghier & Qu, 2013 [124], presented an iterative adaptive approach, that uses a hyper-heuristic to select the best timeslot and room for each exam. The approach consists of three phases. The first phase produces a solution by applying a set of low-level heuristics. The next phase creates a hybridized sequence of heuristics, that will be used in the third phase. In this last phase, the sequence of heuristics is adjusted. The presented model showed good results.

In 2012, Sin & Kham [121] proposed a hyper-heuristic to improve the solution for scheduling exams in the university. The hyper-heuristic used was based on Great Deluge and its variants, Lex Deluge, Nonlinear, and Extended Great Deluge. The method presented the best results for almost all cases, where it was tested. Anwar et al., 2013 [13], presented a harmony search-based hyper-heuristic. The hyper-heuristic applies a high-level heuristic chosen from a set of low-level heuristic methods. In this model, the meta-heuristic Harmony Search is used to get a better space of heuristic sequences, and the feasible solution, found with the largest degree heuristic, is improved with the combination of two-level perturbative heuristics, where the first choose randomly two exams and change the timeslots between them, and the second heuristic is in charge of assign each exam to a different timeslot, having in consideration the hard constraints defined. The obtained results were not the best ones, in comparison with the solutions present in the actual literature.

Pillay & Banzhaf, 2009[99], proposed a different combination of heuristics for Hyper-heuristics systems. Usually, in these types of models heuristics are applied sequentially, however in this study, the researchers applied the combination of heuristics (largest degree, largest weighted degree, largest enrollment, saturation degree, and highest cost heuristic) simultaneously. To facilitate the application of the construction heuristics, conditional and logical operators were used. The quality of the solution obtained with this approach had the same quality as the solutions presented in the literature. Pais & Burke, 2010 [94], proposed a constructive heuristic approach based on Choquet integral, combining, five different heuristics (largest degree, color degree, largest weighted degree, largest enrollment, and saturation degree). Through a fuzzy measure, the importance of the heuristics was obtained, which allows the model to order them. The Choquet integral was used in the model to calculate the difficulty of scheduling a certain exam. Rahman et al., 2014 [106], presented an adaptive approach, that also considers the difficulty of scheduling the exams when the model is finding a solution. This difficulty is measured through modification and low-level heuristics (largest and saturation degree). The solutions found by the model were competitive with the ones presented in the literature, which means the results presented high quality.

Hybrid approaches are also applied to UETP. Qu et al., 2009 [103] proposed a hybrid approach, that uses the largest weighted degree with saturation degree, in different stages of the model. These two approaches were chosen through statistical analysis with random iterative graph-based hyper-heuristics, in a collection of heuristics. Alzaqebah & Abdullah, 2015 [12], proposed a hybrid bee colony optimization method. The bee colony optimization is a heuristic, based on the real behavior of the bees. This heuristics works with forward pass actions, where search space is explored, and backward pass actions, where the information found in the previous action is shared with the others. The researchers applied other local search techniques, such as late acceptance hill climbing and simulated annealing algorithms, to the forward pass action. Besides that, three strategies to pass the information collected were tested in the model: tournament, rank, and disruptive selection strategies. The disruptive selection strategy was the one that proved to be the most efficient. In addition to the changes in the basic bee colony optimization previously mentioned, a new way of selecting a neighborhood structure with a self-adaptive mechanism was implemented.

Application

The models produced by the researchers were tested in real data, obtained from the datasets of certain universities. Some cases tested the models in the ITC instances or the Carter et al. benchmarks.

The information mentioned before about UETP can be consulted, all summarized, in Appendix A, for easy categorization of the analyzed researches, where the types of existing problems, solution approaches presented and their details, and in which data the model was tested, are presented.

- **Thesis defence Timetabling Problem**

TDTP is another type of timetabling problem, less studied by the researchers. This is a problem that some universities face when students submit their thesis, to conclude their degree. After that submission, a schedule of the dates that each student will present and defend their thesis has to be done.

Problem Definition

This problem can be considered an optimization problem. Consists of scheduling the different thesis defences, and allocating them to the available timeslots and rooms. The choice of the jury can vary from university to university, depending on the rules of each institution, for example, the number of elements of the jury can be three (Fastré et al., 2017 [51]), five (Huynh et al., 2012 [61], Dung et al., 2015 [49], six (Kochanikova & Rudova, 2013 [73]), or may even vary within the same institution, not being a pre-established number (Battistutta et al., 2019 [20]). Also, the way the jury is chosen can vary, since in some institutions the supervisor of the student is not allowed to be present in the thesis defence (Huynh et al., 2012 [61]), but in other universities is mandatory to be present (Battistutta et al., 2019 [20]). Another difference that is possibly found in the actual literature available about this theme, is that in some cases the committee is chosen before the scheduling is produced, i.e each committee already is assigned to a thesis defence, before the timetable is done, but in other cases the assigned of the jury members to each defence in only done when the scheduling is established (Huynh et al., 2012 [61]). Due to the larger number of thesies submitted in the same date, this problem becomes very time-consuming and complex for the staff responsible for creating this schedule, which means that everything possible must be done to make this process as efficient as possible.

Problem Constraints

As in the other problems mentioned previously in this literature review, this one also makes a distinction between hard and soft constraints. All the hard constraints defined must be respected to find a feasible solution. The quality of the solution will be greater as the number of constraints not violated. There are multiple constraints that can be mentioned for this type of problem, differing for each university. Therefore, the most common ones will be highlighted:

- R1: The elements of the jury must be all different;
- R2: A chairperson must be choose be to part of the jury;
- R3: The element of the jury can only be present in on defence at the same time;
- R4: A room can only be assigned to one defence at the same time;
- R5: When thesis defences allocated to consecutive timeslots have elements of the jury in common, the distance between the rooms where the event will take place must be minimized;
- R6: The number of elements defined for the composition of the jury must be respected.

Objective Function

The objective function of the Master Thesis Timetabling Problem use to be the minimization of the violation of the soft constraints, which means the solution given by the model is the one where the soft constraints are more respected. Besides that, some models assigned a weight to each constraint, to make a difference in their importance. In this way, not respecting a specific soft constraint has a higher impact than not respecting a different constraint. Most of the available models define a single objective function, however, is also possible to find studies where the objective function is multi objective, for example the study conducted by Dung et al., 2015 [49].

Approaches and Methodologies

Few researchs has been carried out and applied to this type of problem, therefore there is not a higher number of models proposed for the Master Thesis Timetabling Problem, in contrast to what happens with the other problems mentioned above.

Huynh et al., 2012 [61] proposed a genetic algorithm, with multiple objectives, to be applied in a real case, of a university in Vietnam. Four objectives were defined in this model: minimize the number of overlaps and the busy time of professors, maximize the number of professors experts in the theme, present in the jury, maximize the number of consecutive timeslots that each professor is assigned, and minimize the number of times that each professor has change rooms. A fitness function was used to represent each objective function, and crossover and mutation operators were used. Kochanikova & Rudova, 2013 [73], proposed a model for scheduling the Bachelor State Examination. The proposed model is in charge of, in the first phase, creating the commissions, assigning them to a day and to a room, and in the second phase, allocating each student to a timeslot and to a commission. The goal of this study was to change the way this timetabling was done, moving from manual scheduling to automatic scheduling, making the process more efficient. Dung et al., 2015 [49] presented a constraint-based local search. This approach explores partial solution space, using the advantages of Contraint-based, such as compositionality, modularity, and reusability. To solve the problem, a java library, constructed by the researchers was used. Fastré et al., 2017 [51] presented an automated tool for scheduling master thesis defence. Their concern about this subject arose when the complexity of this problem increased, in the Louvain School of Engineering. That happened, because of the expansion of the types of themes of each master thesis, which included themes from other universities. With that arrived the difficulty in the coordination of schedules of the committee members. To solve this problem, two Mixed Integer Programming Models were tested. The first model didn't allow unscheduled thesis, unlike the second model, which allowed some theses to remain unallocated. The second model was the one that shows to be the best in this type of situation. The produced model was responsible for allocating a committee, a timeslot, and a room to a master thesis defence, and the presented solution is a timetable, available to all stakeholders. Battistutta et al., 2019 [20] proposed three different approaches to solve this type of problem, one based on Integer Programming, the other based on Constraint Programming, and the last one based on simulated annealing. The goal of the models is to construct juries to evaluate the master thesis defences and allocate them and students to timeslots. The model allocates juries to thesis defences based on the expertise level of the jury elements in the master's theme that will be evaluated. The three methods were compared and the conclusion was that the best solution was found through the model based on Integer Programming. In 2021, Christopher & Wicaksana [35] proposed a Particle Swarm Optimization Algorithm to schedule master thesis defences at University Multimedia Nusantara, in Indonesia. At this university, the examiner and moderator of the session are defined by the department, which means the model is not responsible for that allocation. The model presented, as a solution, a timetable with twenty-five thesis defences scheduled, without breaking any hard constraints. The approach proved to be more efficient than the method used previously, saving hours of work for the staff who were in charge of this task.

Application

To evaluate the models proposed by the researchers, they applied the approaches to real data, from some universities. This way, they could conclude if the model presented helped the university to turn the process of scheduling the master thesis defences more efficient.

The information mentioned before about TDTP can be consulted, all summarized, in Appendix A, for easy categorization of the analyzed researches, where the solution approaches presented and their details, and in which data the model was tested, are presented.

3.2 Conclusions

Timetabling problems are widely studied problems, for many years, with several works developed in this theme. The problem of resource allocation at a given time is transversal to several areas such as train scheduling, sports events, employee shifts, and academic class scheduling (Cacchiani & Paolo, 2012 [28], Trick, 2000 [137], Meisels & Schaerf, 2003 [83] and Pillay, 2016 [97]). Educational scheduling problems are varied, grouping problems of allocation in schools and universities, with the latter grouping problems UCTP, UETP and TDTP. The first two rely on extensive research, with the development of exact and non-exact models to solve this type of problem, applying both real data and computationally generated data, in order to test the effectiveness of the models developed.

The problem less developed and addressed in the literature is TDTP which has a small number of articles developed by Huynh et al., 2012 [61], Kochanikova & Rudova, 2013 [73] and Christopher & Wicaksana, 2021 [35], with non-exact methods, namely Generic Algorithms, Local Search and Particle Swarm Optimization. The works developed by Dung et al., in 2015 [49], by Fastre et al., in 2017 [51] and Battistutta et al., in 2019 [20] added the application of exact models for the resolution of TDTP. The constraints applied in each model vary according to each case study taken into account. The study by Fastré, et al, in 2017 [51] presents constraints for the definition of each jury of each thesis simultaneously with the allocation of each thesis defence to a day and time. On the other hand, there are models that rely on the establishment of juries prior to the allocation of thesis defences [49]. The concern with obtaining more compact schedules, so that there is not a large distribution of thesis defences for each day, for each member, is also taken into account in the work of Battistutta et al., 2019 [20].

Thus, in order to add value to a less developed branch of research, as is the case of TDTP, an exact model will be developed, for the specific case of the allocation of thesis defences in DEG, according to its needs and constraints. It should be noted that none of the models fully translates the allocation process of MEGI thesis defenses into DEG, so their construction will add value to the existing literature.

4. Mathematical Model

The present chapter 4 will present two main themes, the problem description and the problem formulation. The former expounds on the case study of the DEG at IST, as set forth in section 4.1. The second one, section 4.2, presents the constraints and objectives that the department has in consideration when is doing the schedule of the master thesis defences. Finally, the conclusions of the chapter will be presented, in section 4.3.

4.1 Problem Description

As previously mentioned, the allocation of thesis defences is a problem faced by many universities, requiring significant effort and resources to be resolved. This study focuses on the specific case of DEG at IST, where the department's secretary manually performs the timetabling of thesis defences. The scheduling process involves contacting members of each jury by email or phone to identify an available date that suits all involved parties. This manual process is time-consuming, taking up to a month to schedule a small number of thesis defences. The problem has several challenges, but the availability of jury members and the number of thesis to schedule were identified as the most critical factors contributing to its complexity.

As stated before, the composition of each thesis defence jury includes three professors: the chairperson, who typically serves as a member of the jury in a larger number of thesis defences, the supervisor who guides the student throughout the research and development of the thesis, and an additional member.

The availability of jury members is often limited by their other responsibilities as professors in the university. In addition, the number of thesis defences to be scheduled varies depending on the number of students submitting their thesis within a given period. As a result, the difficulty of scheduling thesis defences increases during the months of May and October, which are the deadlines for thesis submission and typically have a higher number of submissions.

Furthermore, it is worth noting that in addition to the aforementioned factors, the geographic distribution of the faculty across the two campi of IST, namely Alameda and Taguspark, adds another layer of complexity to the scheduling process. Specifically, it is desirable that professors who are members of multiple thesis defence committees have their commitments compacted into the fewest number of days possible, while taking into account their availability, in order to minimize the need for travel between the two campi.

Therefore, considering the constraints and objectives of the department, the mathematical model was designed to assign a specific day and time for each thesis defence. The availability and weight of each member of the jury were then taken into account to allocate them to a reduced number of days and spaced timeslots, minimizing the need for them to be present on multiple days and avoiding schedule conflicts.

4.2 Problem Formulation

To build the model, the concerns of the DEG were taken into account, reflected in the constraints and objectives defined in it. Three types of constraints where considered, the Structural Constraints, the Compactness Constraints and the Professors Constraints. In the existent literature, is possible find other constraints used in different models, for example, models defined in order to be able to form the jury of each thesis and to have in consideration the distance of each room, in the case of universities with different campi (Huynh et al. 2012 [61], Dunget al., 2015 [49], Christopher & Wicaksana, 2021 [35]), as mentioned before. Since the case at hand does not have this type of limitations, once the juries of each thesis are formed before by the scientific committee of the department, and even though the university has two campi, the defences of the DEG are all placed in Campus Taguspark.

In some instances in the literature, the number of available rooms for thesis defences has been taken into consideration by assigning a room to each defence in the model. However, in the case of DEG at IST, the allocation of rooms is done in a next stage and is not the responsibility of the department. Therefore, this will not be considered as a constraint in the mathematical model developed.

This section presents all the sets and parameters used to define the Mixed Integer Programming Model built, as well as the decision variables.

A set I is defined, comprising the thesis defences that need to be scheduled. These thesis defences must be assigned a specific day $d \in D$ and a corresponding timeslot $t \in T$, while keeping in mind the 5 working days of the week. Each working day is composed of 7 distinct timeslots, which must be respected to account for the duration of each thesis defence.

In the current context, the eligible members who are capable of performing the roles of chairperson, supervisor, and additional member for each thesis defence are defined in the set M. The members assigned to the defence of each student are established using the J_{in} parameter, as the jury of each thesis is defined before scheduling the defence. Each committee is formed by an NJ number of people. The availability of the professors varies due to differences in their teaching loads, roles, and campi locations. Therefore, their availability is collected over the days and timeslots considered for the defences and is presented by the parameter A_{mdt}. The availability of member $m \in M$ is represented by 1 if they are available at day $d \in D$ and timeslot $t \in T$ and 0 if they are not.

The responsibility of allocating rooms for the thesis defence does not fall on the department. However, the mathematical model takes into account the R parameter, which represents the number of rooms available on the Taguspark campus that must be respected for each timeslot of each day.

Given that the chairpersons of each jury are, as a rule, the professors with the lowest availability, belonging to a higher number of committees, it is necessary to define different weights for each member according to the role played. The weights assigned to each member of the jury, according to the role played, are defined by the parameters $W1_m$ and $W2_m$. $W1_m$ assigns weights based on the penalty for the increasing number of days that each member has to be present in thesis defences, while $W2_m$ assigns weights based on the number of gaps between thesis defences that each member has in their schedule. These weights are represented by integer values, where higher values indicate a higher penalty.

The main objective of this model is to allocate all thesis defences to one day $d \in D$ and timeslot $t \in T$. For this reason it is necessary to use binary decision variable X_{idt}, which takes the value of 1, if the thesis $i \in I$ is allocated to day $d \in D$ and timeslot $t \in T$. Otherwise, the variable takes the value of zero.

In order to optimize the available time of the members of the jury and to respect their preferences, it was necessary to define constraints that would guarantee that the thesis defences of each member, represented by TJ_{im}, were allocated to the fewest number of days and with the smallest spacing of hours possible, in order to make the schedules compact for each member. Therefore, it was necessary to use auxiliary variables, such as the variable S_{md} which returns the value 1 if the member $m \in M$ has thesis defences allocated to day $d \in D$ and the variable G_m, which counts the number of days that member $m \in M$ has at least one thesis defence allocated. In order to count the days of each member, another auxiliary variable was created, GQ_{mq} which takes the value of 1 when member $m \in M$ has theses allocated in the number of days $q \in Q$. The set Q combines all possible values of being the maximum number of days that a professor has to be a member of a thesis jury, that is, it combines all values from 0 to B, where B is the maximum number of days that the secretary the department has available to do the scheduling. All the previously mentioned auxiliary variables are used in the constraints that minimize the number of days a member has to attend a thesis defence.

In addition, variables are also used for maximizing the compactness of the schedule, such as Y_{md} which represents the first timeslot of day $d \in D$ that professor $m \in M$ has a thesis defence. In turn, the variable Z_{md} represents the last timeslot in which member $m \in M$ has to attend a thesis defence on day $d \in D$. To account for the number of empty timeslots between scheduled theses, of each member $m \in M$ on day $d \in D$, the variable H_{md} is used, which takes into account the values of Y_{md} and Z_{md}.

The indices, sets, subsets, parameters and variables used in the mathematical model are summarized in Table 4.1.

The implementation of the model will be addressed in the following subsections, through the construction of four types of constraints: Structural, compactness and professors constraints, in subsections 4.2.1, 4.2.2, 4.2.3 and 4.2.4 respectively.

Table 4.1: Indices, sets, subsets, parameters and variables for the mathematical model

Indices and Sets	
$i \in$ I	Set of master thesis defences
$m \in$ M	Set of jury members
$d \in$ D	Set of available days
$t \in$ T	Set of available timeslots
$n \in$ N	Set of possible positions of a jury member
$q \in$ Q	Set of possible number of days jury members have defences on
Parameters	
J_{in}	Jury composition of each master thesis defence
TJ_{im}	1 if member m belongs to the jury of thesis i defence, 0 otherwise
R_{dt}	Number of rooms available each day d in timeslot t
A_{mdt}	1 if member m is available at day d and timeslot t, 0 otherwise
$W1_m$	weight of schedule member m among different days
$W2_m$	weight of number of holes in the schedule of member m
EXP	Exponential penalty
B	Maximum number of days a committee member can be scheduled for
NT	Number of thesis to schedule
Auxiliary variables	
S_{md}	1 if member m has thesis defence in day d, 0 otherwise
G_m	Number of days a committee member m has a defence in
GQ_{mq}	1 if member m has thesis defences scheduled in q days, 0 otherwise
Y_{md}	First timeslot of day d that member m has thesis defences scheduled on
Z_{md}	Last timeslot of day d that member m has thesis defences scheduled on
H_{md}	Number of holes that member m has in the schedule of day d
Decision Variable	
x_{idt}	1 if thesis i is scheduled in day d and timeslot t, 0 otherwise

4.2.1 Structural Constraints

This subsection describes the constraints that must be satisfied, defined as hard constraints. The model establishes that each thesis defence $i \in I$ must be allocated to only one timeslot (4.1).

$$\sum_{d\in D}\sum_{t\in T} x_{idt} = 1, \forall i \in I \tag{4.1}$$

In order to respect the defined jury of each thesis defence and the availability of each member, two constraints were defined, which guarantee that each member of the jury of thesis $i \in I$, J_{in}, is available on day $d \in D$ e timeslot $t \in T$ to which the thesis defence was allocated (4.2) and that members of thesis defences juries cannot have two thesis defences scheduled for the same timeslot (4.3), i.e. that the sum of all thesis defences allocated in day $d \in D$ and timeslot $t \in T$, where member $m \in M$ belongs to the jury, must be equal or less than 1.

$$x_{idt} \leq A_{J_{in}dt}, \forall i \in I, \forall n \in N, \forall d \in D, \forall t \in T \tag{4.2}$$

$$\sum_{i\in I:TJ_{im}=1} x_{idt} \leq 1, \forall d \in D, \forall t \in T, \forall m \in M \tag{4.3}$$

Although the definition of rooms for each thesis defence is not up to the department, it is important to take into account the number of theses allocated to the same $t \in T$ and $d \in D$, so that the total does not exceed the number of classrooms available on campus. Thus, a constraint was established, to ensure that the sum of all thesis defences allocated in the same day and timeslot is less or equal to the number of rooms available in the same day and timeslot, R_{dt} (4.4).

$$\sum_{i\in I} x_{idt} \leq R_{dt}, \forall d \in D, \forall t \in T \tag{4.4}$$

4.2.2 Compactness Constraints

Compactness constraints use auxiliary variables in order to minimize the number of empty slots between thesis defences allocated in each day, which the member has to attend, so that there is not a large dispersion, for each member, of the public evidence throughout the day. The values of the variables S_{md}, Y_{md}, Z_{md} and H_{md} are defined.

These constraints are considered soft constraints, since there is no obligation associated with each one of them, not being associated with the feasibility of the model. These only establish values of the auxiliary variables used later in the minimization of the objective function, that is, in this case the number of holes, H_{md}, will be determined between each defence $i \in I$ allocated to each day $d \in D$, of each member $m \in M$, and will be used in the objective function in order to obtain the smallest possible value.

Thus, it was necessary to use an auxiliary variable S_{md} to determine whether member $m \in M$ has to attend thesis defences on day $d \in D$. This variable assumes the value of 1 if member $m \in M$ has thesis defences on day $d \in D$, and 0 otherwise. Two constraints were used for this purpose (4.5) and (4.6), the first of which establishes that the variable will have a value of 1 if there is at least one thesis allocated to day $d \in D$, within the theses to which the professor is allocated as belonging to the jury. In turn, the second constraint guarantees that if no thesis of the member is allocated to day d, the auxiliary variable will take the value of 0.

$$S_{md} \geq x_{idt}, \forall m \in M, \forall d \in D, \forall t \in T, \forall i \in I : TJ_{im} = 1 \tag{4.5}$$

$$S_{md} \leq \sum_{t \in T} \sum_{i \in I : TJ_{im}=1} x_{idt}, \forall m \in M, \forall d \in D \tag{4.6}$$

To calculate the value of the auxiliary variable, to be used in the objective function, H_{md}, three constraints (4.7) to (4.9) were necessary. The first two define the first and last timeslot of each day in which member $m \in M$ has a thesis defence allocated, and if none of the theses that the professor is defined as jury is allocated to day d, the variable auxiliary Y_{md}, which fixes the first timeslot in which member m has a thesis defence on day $d \in D$, will take the value of NT, total number of theses to allocate, plus 1. This way, when calculating the number of holes, on each day, between the thesis defences of each member (4.9), the auxiliary variable H_{md} will take the value of 0. On the other hand, if member $m \in M$ has thesis defences on day $d \in D$, the number of holes will be the difference between the last and the first timeslot of the day with thesis defences allocated, minus the total thesis defences allocated throughout the day. The variable H_{md} will be used in the objective function, in order to minimize it, ensuring that the number of empty slots between each thesis defence is as small as possible, guaranteeing more compact schedules for each member of the jury.

$$Y_{md} \leq (NT + 1) - (NT + 1 - t) \cdot \sum_{i \in I : TJ_{im}=1} x_{idt}, \forall m \in M, \forall d \in D, \forall t \in T \tag{4.7}$$

$$Z_{md} \geq t \cdot \sum_{i \in I : TJ_{im}=1} x_{idt}, \forall m \in M, \forall d \in D, \forall t \in T \tag{4.8}$$

$$H_{md} \geq Z_{md} - Y_{md} + S_{md} - \sum_{i \in I : TJ_{im}=1} x_{idt}, \forall m \in M, \forall d \in D \tag{4.9}$$

4.2.3 Professors Constraints

Professors have different roles and responsibilities, which can sometimes be divided between the two university campi. Therefore, it is essential that the defences to which they are allocated are not dispersed over a large number of days, being important to minimize this number. Once again, these constraints are soft constraints, and there is no obligation associated with them. They only have the objective of making the allocation of thesis defences fairer for each member, taking into account the different positions and preferences of each one. Thus, these auxiliary variables are used in the objective function, as will be demonstrated in subsection 4.2.5, in order to maintain its linearity.

For this purpose, four constraints were defined (4.10) to (4.13), three auxiliary variables were used, S_{md}, a binary variable that defines whether member $m \in M$ has thesis defences on day $d \in D$, G_m, which determines the total number of days that member $m \in M$ has to participate in thesis defences, and GQ_{mq}, again a binary variable, to determine whether member $m \in M$ has defences in the total number of days $q \in Q$. Two parameters B and Q were used.

The first two constraints (4.12) and (4.13) determine the total number of days in which each member $m \in M$ has at least one thesis defence allocated, being mandatory their presence at Campus Taguspark in that number of days. The first constraint determines the total number of days through the sum of the binary variable S_{md} defined in the compactness constraints, and the second constraint establishes an upper bound for the variable.

$$\sum_{d \in D} S_{md} = G_m, \forall m \in M \tag{4.10}$$

$$G_m \leq B, \forall m \in M \tag{4.11}$$

For later use in the objective function, it was also necessary to define the binary variable GQ_{mq} which takes a value of 1 when member $m \in M$ has thesis defences allocated in the total number of days $q \in Q$, returning 0 to all others. This variable allow assigning an exponentially increasing weight as greater the number of days which member $m \in M$ is allocated. The variable GQ_{mq} was defined with constraints (4.13) and (4.14). The definition of this auxiliary variable guarantees the linearity of the objective function.

$$\sum_{q \in Q} GQ_{mq} = 1, \forall m \in M \tag{4.12}$$

$$\sum_{q \in Q} GQ_{mq} \cdot q = G_m, \forall m \in M \tag{4.13}$$

4.2.4 Domain Constraints

To guarantee obtaining variables within the required standards, it was necessary to build constraints that established the domains of the various variables defined and used in the development of the model, that is, domain constraints for the variables S_{md}, G_m, GQ_{mq}, Y_{md}, Z_{md}, H_{md} and x_{idt}, (4.14) to (4.20) The variables S_{md}, GQ_{mq} and x_{idt} are binomial variables that can take the values 0 and 1, the rest being positive integer variables.

$$S_{md} \in \{0,1\}, \forall m \in M, \forall d \in D \tag{4.14}$$

$$G_m \in \mathbb{Z}_0^+, \forall m \in M \tag{4.15}$$

$$GQ_{mq} \in \{0,1\}, \forall m \in M, \forall q \in Q \tag{4.16}$$

$$Y_{md} \in \mathbb{Z}_0^+, \forall m \in M, \forall d \in D \tag{4.17}$$

$$Z_{md} \in \mathbb{Z}_0^+, \forall m \in M, \forall d \in D \tag{4.18}$$

$$H_{md} \in \mathbb{Z}_0^+, \forall m \in M, \forall d \in D \tag{4.19}$$

$$x_{idt} \in \{0,1\}, \forall i \in I, \forall d \in D, \forall t \in T \tag{4.20}$$

4.2.5 Objective Function

The mathematical model developed has two components to be optimized, being the objective function the sum of them, so that the allocations of thesis defences are established taking into account the preferences and responsibilities of each member of the jury. Thus, the first component defined was the minimization of the total number of days in which each member has thesis defences allocated (4.21). This was defined to attribute an exponentially increasing weight as greater the total number of days in which the member $m \in M$ has to attend a thesis defence. Since there are professors with a greater number of responsibilities, less availability, and who have roles on different campi, different weights are assigned to each member. The higher the level of responsibilities and occupation, the greater the weight assigned. This weight is defined by the person responsible for allocating the thesis defences, with the assignment of different weights to each member being at their discretion. In the case study to be analyzed, this task would be in charge of the secretary of DEG.

$$\sum_{m \in M} \sum_{q \in Q} (W1_m \cdot GQ_{mq} \cdot q^{EXP}) \tag{4.21}$$

The second component of the objective function is to minimize the number of holes between each thesis defences that each member has to attend, in each day. In the same way as presented in the previous component, a different weight is assigned to each member, leaving the responsibility to the decision maker of making this decision, according to the responsibilities of each member. The greater the assigned weight and the greater the number of holes between thesis defences allocated in the same day, the greater the assigned penalty.

$$\sum_{m \in M} \sum_{d \in D} (W2_m \cdot H_{md}) \tag{4.22}$$

Overall, taking into account the aforementioned components, the objective function of the model is defined as the minimization the sum of the two components (4.23).

$$min(\sum_{m \in M} \sum_{q \in Q} (W1_m \cdot GQ_{mq} \cdot q^{EXP}) + \sum_{m \in M} \sum_{d \in D} (W2_m \cdot H_{md})) \tag{4.23}$$

4.3 Conclusions

This chapter describes the mathematical model developed to carry out the scheduling of thesis defences of DEG of IST, as an alternative to the current, less efficient method, which allocates them manually, through the department secretary. It is a problem present in the literature, with some previously published models (Huynh et al., 2012 [61], Kochanikova & Rudov, 2013 [73], Dung et al., 2015 [49], Fastré et al., 2017 [51], Battistutta et al., 2019 [20], Almeida, 2021 [8]), but with different characteristics from those considered for DEG.

The model has the objective of minimizing the total number of days that each member has to be present in at least one thesis defence, and the minimization of empty slots between each thesis defence, present on the same day, for each member. Therefore, the objective of the model is the allocation of thesis defences in the most compact and efficient way for each professor.

The model was defined as a Mixed Integer Programming Model, which is an exact approach, that achieves feasible or infeasible solutions, depending on the size of the instances, defined by the total number of thesis defences to be allocated, the constitution of each thesis defence jury and the availability of each member. Since these are the conditions that influence the feasibility of each solution, it is important to analyze the possibility of optimizing the scheduling of thesis defences, by changing one of these factors, as will be proposed in the next chapter.

5. Alternative Mathematical Model

This chapter consists of a section with the description of an alternative model to the one presented above, where parameters, indices, variables, constraints and objective function of the model, necessary for its construction, are defined. Section 5.1 presents the problem formulation and Section 5.2 presents the conclusions of the chapter.

5.1 Problem Formulation

In order to optimize the way in which the allocation of thesis defences is carried out by the department, an alternative model was built, based on the previously demonstrated model. The new model intends to test the hypothesis that the allocation is more effective if the formation of thesis defence juries is carried out in parallel with the allocation of each thesis defence, that is, the definition of each jury is in charge of the model. In this way, it is expected that the schedules obtained will be carried out in a better way for each member, not requiring so many days to carry out all the thesis defences. In the existing literature, it is possible to find works that use this type of constraints for the development of the model, in order to elaborate the definition of the juries of each thesis defence (Huynh et al. 2012 [61]).

The new model also presents four sets of types of constraints, the structural constraints, the compactness constraints, the professors constraints and the domain constraints, and in the following subsections the changes made are presented, so that the model is able to define the juries of each thesis, and allocate them to different days and timeslots, according to their availability and position.

For the definition of the new model, it is no longer necessary to use the parameters J_{in} and TJ_{im} since the committees are no longer constituted prior to the scheduling of thesis defences. Contrary, new parameters were added, such as SU_i, which corresponds to the supervisor of each thesis, SUP_{im}, which takes the values of 1 if member $m \in M$ is the supervisor of thesis $i \in I$, and 0 otherwise, and the parameter CH_m, in order to characterize each member according to their ability to be chairperson of a committee, returning the value of 1 if member $m \in M$ can be considered for the position of chairperson. In addition to the changes described in the parameters, a new binary decision variable was also added, w_{imdt}, which has a value of 1 if thesis $i \in I$ and member m are allocated to the same day $d \in D$ and timeslot $t \in T$, and 0 otherwise . In this way, the model will not only allocate each thesis to a timeslot, but also the members that will constitute the jury of each thesis defence.

The indices, sets, subsets, parameters and variables used in the mathematical model are summarized in Table 5.1.

The implementation of the model will be addressed in the following subsections, through the construction of four types of constraints: Structural, compactness, professors and domain constraints, in subsections 5.1.1, 5.1.2, 5.1.3 and 5.1.4 respectively.

5.1.1 Structural Constraints

Structural constraints follow the same principle as those defined in the previously described model, with the mandatory allocation of all thesis defences to one day and timeslot (5.1), with no members having more than one thesis allocated to the same day $d \in D$ and timeslot $t \in T$ (5.2), in addition to the fact that each member can only be allocated to that schedule if the member is available for that time (5.3).

$$\sum_{d \in D} \sum_{t \in T} x_{idt} = 1, \forall i \in I \tag{5.1}$$

$$\sum_{i \in I} w_{imdt} \leq 1, \forall_{m \in M}, \forall_{d \in D}, \forall_{t \in T} \tag{5.2}$$

$$w_{imdt} \leq A_{mdt}, \forall i \in I, \forall m \in M, \forall d \in D, \forall t \in T \tag{5.3}$$

The changes can be seen in constraints (5.4) to (5.7), which were defined to enable the model to choose the committee for each thesis defence, allocating 3 elements, a chairperson, the supervisor and an additional member at the same day and timeslot. Thus, it is necessary to define a constraint that guarantees that the supervisor of each thesis is allocated at the same scheduled time for its defence (5.4).

$$x_{idt} = w_{imdt}, \forall i \in I, \forall d \in D, \forall t \in T, SUP_{im} = 1 \tag{5.4}$$

Since the presence of a chairperson is required in each thesis defence, and taking into account that not all professors are suitable for this position, two constraints were defined to respected this criterion (5.5) and (5.6). Since the supervisor of each thesis cannot play the role of chairperson, even if the professor is capable of performing the role, it is necessary to allocate a second element for this purpose. In this way, the sum of all elements capable of playing the role of chairperson, allocated to day $d \in D$, timeslot $t \in T$, and associated with the committee of thesis $i \in I$ must be equal to 1 (5.5), in case the supervisor of thesis i is not able to to be a chairperson, or 2 otherwise (5.6).

Table 5.1: Indices, sets, subsets, parameters and variables for the mathematical model

Indices and Sets	
$i \in$ I	Set of master thesis defences
$m \in$ M	Set of jury members
$d \in$ D	Set of available days
$t \in$ T	Set of available timeslots
$n \in$ N	Set of possible positions of a jury member
$q \in$ Q	Set of possible number of days jury members have defences on
Parameters	
R_{dt}	Number of rooms available each day d in timeslot t
A_{mdt}	1 if member m is available at day d and timeslot t, 0 otherwise
$W1_m$	weight of schedule member m among different days
$W2_m$	weight of number of holes in the schedule of member m
SU_i	supervisor of thesis i
SUP_{im}	1 if member m is the supervisor of thesis i, 0 otherwise
CH_m	1 if member m can be chairman, 0 otherwise
EXP	Exponential penalty
B	Maximum number of days a committee member can be scheduled for
NT	Number of thesis to schedule
Auxiliary variables	
S_{md}	1 if member m has thesis defence in day d, 0 otherwise
G_m	Number of days a committee member m has a defence in
GQ_{mq}	1 if member m has thesis defences scheduled in q days, 0 otherwise
Y_{md}	First timeslot of day d that member m has thesis defences scheduled on
Z_{md}	Last timeslot of day d that member m has thesis defences scheduled on
H_{md}	Number of holes that member m has in the schedule of day d
Decision Variable	
x_{idt}	1 if thesis i is scheduled in day d and timeslot t, 0 otherwise
w_{imdt}	1 if thesis i is scheduled in day d and timeslot t, 0 otherwise

$$\sum_{m \in M : CH_m = 1} w_{imdt} = x_{idt}, \forall i \in I : CH_{SU_i} \neq 1, \forall d \in D, \forall t \in T \tag{5.5}$$

$$\sum_{m \in M : CH_m = 1} w_{imdt} = 2 \cdot x_{idt}, \forall i \in I : CH_{SU_i} = 1, \forall d \in D, \forall t \in T \tag{5.6}$$

After allocating the supervisor of each thesis at the same time and defining a chairperson for each thesis defence, it is also necessary to allocate a third element. Thus, the sum of the total elements allocated to the same time of each thesis defence will be equal to 3, ensuring that all allocated members are different (5.7)

$$\sum_{m \in M} w_{imdt} = 3 \cdot x_{idt}, \forall i \in I, \forall d \in D, \forall t \in T \tag{5.7}$$

5.1.2 Compactness Constraints

As the model developed and presented previously, the new model aims to build compact schedules for each member, in order to minimize the number of empty slots between each thesis defence to which they are allocated per day. Therefore, five constraints were also built to guarantee the definition of the auxiliary variable H_{md} to be used later in the objective function. The defined constraints have the same function as those defined in the previous model, that is, define the total number of days in which member $m \in M$ has thesis defences allocated (5.8) and (5.9), determine the first and last timeslot of each day in that each member has to attend a thesis defence, (5.10) and (5.11), and define the total number of holes that each member has in each day (5.12). All constraints follow the same line of thought as the constraints defined for the first model, with the differentiation of the decision variable used in their construction. In the new model the decision variable w_{imdt} is used.

$$S_{md} \leq \sum_{t \in T} \sum_{i \in I} w_{imdt}, \forall m \in M, \forall d \in D \tag{5.8}$$

$$S_{md} \geq w_{imdt}, \forall m \in M, \forall d \in D, \forall t \in T, \forall i \in I \tag{5.9}$$

$$Y_{md} \leq (NT + 1) - (NT + 1 - t) \cdot w_{imdt}, \forall m \in M, \forall d \in D, \forall t \in T \tag{5.10}$$

$$Z_{md} \geq t \cdot w_{imdt}, \forall m \in M, \forall d \in D, \forall t \in T, \forall i \in I \tag{5.11}$$

$$H_{md} \geq Z_{md} - Y_{md} + S_{md} - \sum_{i \in I} \sum_{t \in T} w_{imdt}, \forall m \in M, \forall d \in D \tag{5.12}$$

5.1.3 Professors Constraints

The new model has the same constraints that the previous model used to make it possible to define the auxiliary variable GQ_{mq}, to be used in the objective function. The four constraints define the total number of days in which member $m \in M$ has thesis defences (5.13) and (5.14) and define the auxiliary binary variable to be used in the objective function that presents the value 1 if member $m \in M$ has thesis defences allocated on $q \in Q$ days, 0 otherwise, (5.15) and (5.16).

$$\sum_{d \in D} S_{md} = G_m, \forall m \in M \tag{5.13}$$

$$G_m \leq B, \forall m \in M \tag{5.14}$$

$$\sum_{q \in Q} GQ_{mq} = 1, \forall m \in M \tag{5.15}$$

$$\sum_{q \in Q} GQ_{mq} \cdot q = G_m, \forall m \in M \tag{5.16}$$

5.1.4 Domain Constraints

To guarantee obtaining variables within the required standards, it was necessary to build constraints that established the domains of the various variables defined and used in the development of the model, that is, domain constraints for the variables S_{md}, G_m, GQ_{mq}, Y_{md}, Z_{md}, H_{md}, x_{idt} and w_{idt}, (5.17) to (5.24) The variables S_{md}, GQ_{mq}, x_{idt} and w_{idt} are binomial variables that can take the values 0 and 1, the rest being positive integer variables.

$$S_{md} \in \{0, 1\}, \forall m \in M, \forall d \in D \tag{5.17}$$

$$G_m \in \mathbb{Z}_0^+, \forall m \in M \tag{5.18}$$

$$GQ_{mq} \in \{0, 1\}, \forall m \in M, \forall q \in Q \tag{5.19}$$

$$Y_{md} \in \mathbb{Z}_0^+ \forall m \in M, \forall d \in D \tag{5.20}$$

$$Z_{md} \in \mathbb{Z}_0^+, \forall m \in M, \forall d \in D \tag{5.21}$$

$$H_{md} \in \mathbb{Z}_0^+, \forall m \in M, \forall d \in D \tag{5.22}$$

$$x_{idt} \in \{0,1\}, \forall i \in I, \forall d \in D, \forall t \in T \tag{5.23}$$

$$w_{imdt} \in \{0,1\}, \forall i \in I, \forall m \in M, \forall d \in D, \forall t \in T \tag{5.24}$$

5.1.5 Objective Function

The objective function of the new model were defined in the same way as the objective function of the first model, minimization of the number of days in which each member has thesis defences, and also minimizing the number of empty slots present in each day in which professors have to attend thesis defences, in order to obtain more compact timetables, (5.25) and (5.26).

$$\sum_{m \in M} \sum_{q \in Q} (W1_m \cdot GQ_{mq} \cdot q^{EXP}) \tag{5.25}$$

$$\sum_{m \in M} \sum_{d \in D} (W2_m \cdot H_{md}) \tag{5.26}$$

Overall, taking into account the aforementioned components, the objective function of the model is defined as the minimization the sum of the two components (5.27).

$$min(\sum_{m \in M} \sum_{q \in Q} (W1_m \cdot GQ_{mq} \cdot q^{EXP}) + \sum_{m \in M} \sum_{d \in D} (W2_m \cdot H_{md})) \tag{5.27}$$

5.2 Conclusions

The new model for allocating juries to thesis defences is based on the model described in chapter 4, with an important change. Instead of defining the juries beforehand, as in the previous model, in this new mathematical model the juries are determined simultaneously with the allocation of each thesis defence on a specific day and timeslot. This means that the definition of juries is integrated into the defence allocation process, unlike the previous model where the juries were predefined. This is one of the characteristics present in the model published by Battistutta et al., in 2019 [20].

The new model proposed in this study is more complex than its predecessor, as it takes into account a greater number of constraints and variables, resulting in an increased number of potential solutions and a longer time to obtain the optimal solution.

Furthermore, this new model provides the opportunity to investigate the impact of the way of defining the jury for each thesis defence on the obtained solution. Specifically, it will enable to examine whether defining the jury prior to or in parallel with the allocation of each thesis defence changes the solution obtained, potentially improving the objective function's outcome. The testing of this hypothesis will be carried out in Chapter 6.

6. Results

The problem of scheduling DEG's thesis defences, in IST, was addressed through an exact model, as demonstrated in chapter 4. In an attempt to evaluate and find better solutions, changing the composition of the juries of each thesis defence, a second model was built, described in chapter 5.

Chapter 6 provides a analysis of the results obtained from the two models. Section 6.1 describes the instances used to test the models. Section 6.2 presents the results obtained from the exact model, which represents the current scheduling process of thesis defences in the department. Section 6.3 presents the results of the alternative model proposed in Chapter 5. A comparison of the results obtained using both models is presented in Section 6.4. Finally, Section 6.5 presents the conclusion of the Chapter 6.

To conduct the exact computational experiments in this study, the IBM ILOG CPLEX Optimization Studio was utilized. The two models were implemented in OPL, which is an Optimization Programming Language capable of translating mathematical models, such as linear and mixed-integer programming models. All the computational experiments were executed on a computer equipped with an Intel(R) Core(TM) i7 processor - 7500U CPU @ 2.70 GHz @ 2.90GHz, with 12GB of RAM.

6.1 Instances

In this study, an instance generator was employed to simulate data reflecting the situation observed in DEG during the allocation of thesis defences each semester. The generator created random data for the model, based on the necessary sets. The generator created four parameters, namely the composition of the jury for each thesis defence, J_{in}, the availability of each member, A_{mdt}, and the weights associated with each professor, $W1_m$ and $W2_m$ to allocate penalties according to their positions.

The instances were categorized into 3 groups based on the number of thesis defences necessary to allocated, 96, 51, and 33. The instance generator represented the hypothetical case of all 96 students enrolled in the 2nd semester of the 2020/2021 academic year defending their thesis between October and December. It was also considered the number of students who defended their dissertation in the 2nd semester of the academic year 2019/2020, which was 51. Finally, the generator considered the possibility that only one-third of the enrolled students would defend their thesis, 33 students, in the same semester. The instance generator used was similar to the one developed by Almeida, in 2021 [8] for his dissertation on the same topic, the scheduling of master's thesis defences of DEG.

Each committee consists of a chairperson, supervisor and an additional member. Normally, the number of thesis defences that each member, in each position, has to attend is uniformly distributed, with some exceptions. Therefore, the instance generator represents the allocation of thesis defences to each member through a normal distribution, in order to simulate this behavior. Thus, it was necessary to define the number of thesis defences to allocate and the number of members available for each position. It should be noted that chairpersons can occupy any position on the committee, supervisors can also occupy the position of the additional professor, and the additional member can only occupy that position. The instance generator also ensured that no member was repeated in each committee.

To obtain the availability of each member for each timeslot and day, some changes were made compared to the base instance generator. Three types of instances were generated with different percentages of availability, considering the values of 0.3, 0.5 and 0.7 for the different percentages of availability of each member. In this way, availability values were randomly generated for each day, for the different members of the jury, respecting the different percentages.

Finally, in order to add a penalty in the objective function according to the increasing number of holes and days that each professor has to attend thesis defences, random weight values were generated for each professor. This assignment had some changes compared to the instance generator used in the research developed by Almeida, in 2021 [8]. In this research, a distinction was made between members who can only be the additional member represented by NO, those who can occupy the position of additional member and supervisor, NS, and those who can occupy any of the three committee positions, NC. Thus, each member of each group of positions goes through a Bernoulli trial with probability p of being assigned a weight $W1_m$ in the amount of 6 if $m \in NC$, 5 otherwise, 4 if $m \in NS$, 3 otherwise, and 2 if $m \in NO$, 1 otherwise. For each instance, a probability p of 30% was established. Likewise, $W2_m$ weights were established, with twice the penalties for each member of each group. The purpose of the generation of these values having been defined in this way was to prioritize the minimization of holes and days of members belonging to Nc, due to the high number of positions and functions that they carry out at the university. In addition, priority was also given to minimizing the number of gaps between the allocation of each thesis defence on each day, prioritizing compact schedules for each professor.

The characteristics of each instance generated and later used to test the constructed models are summarized in Table 6.1. It is possible to verify the number of thesis defences to be allocated, the number of professors in each position, the probability of assigning each weight and the different percentages of availability considered.

Table 6.1: Summary of the different parameters of the generated instances

Group	Subgroup	Number of thesis defences	Jury members				
			Chairpersons	Supervisors	Additional Members	Availability	Weights Probability
A	A1	33	10	25	15	0.3	0.3
	A2	33	10	25	15	0.5	0.3
	A3	33	10	25	15	0.7	0.3
B	B1	51	10	25	15	0.3	0.3
	B2	51	10	25	15	0.5	0.3
	B3	51	10	25	15	0.7	0.3
C	C1	96	10	25	15	0.3	0.3
	C2	96	10	25	15	0.5	0.3
	C3	96	10	25	15	0.7	0.3

6.2 Exact Model

Section 6.2 of this report presents the outcomes derived from the exact model employed to schedule thesis defences in the DEG. For each subgroup, three distinct instances were formulated, incorporating the features explained previously. Subsequently, Section 6.2.1 showcases the count of thesis defences that can be assigned, corresponding to instances that do not have a feasible solution based on the mathematical model developed.

Due to the high complexity of the problem, characterized by a large number of variables and constraints, the exact model developed is expected to provide feasible solutions for each instance. However, the computational time required for the model to converge to the optimal solution is directly proportional to the complexity of the problem. To address this issue, a time limit of 10800.00 seconds was set for the model. Therefore, the algorithm terminates either when it finds the optimal solution within the given time limit or when the time limit is exceeded.

The Table 6.2 summarizes the results obtained for the 27 instances generated, grouped into subgroups depending on the number of thesis defences that each one has to allocate, and on the availability of members belonging to the juries of each defence. In addition to the subgroup, the number of thesis defences to allocate and the availability of members for each instance, the table also summarizes computational time values, in seconds, the gap, in %, and the value obtained for the objective function, of each generated instance. The measured gap is the percentage difference between the best integer solution and the best node found, considering that the best node is found by the model through decisions made in its search tree, to explore different feasible solutions. If the algorithm manages to find a feasible solution, within the computational time limit, the displayed gap is 0%.

Upon analyzing Table 6.2, it is evident that feasible solutions were not found for instances belonging to subgroups A1, B1, and C1, which have a common jury member availability of 30%. The model has defined, as a hard constraint, the obligation to allocate all the thesis defences. Therefore, based on the fact that the committees for each thesis defence are predefined, if there is no schedule available for all members, the solution will be infeasible. It is possible to directly relate the low percentage of availability of the members with the fact that feasible solutions were not found. In turn, the remaining instances, belonging to subgroups A2, A3, B2, B3, C2 and C3 obtained feasible solutions. However, it should be noted that not all solutions are optimal, only the solutions found for subgroups A2, A3 and B2, through an average computational time of 30.65s, 3248.78s and 1188.78s, respectively. We can draw the aforementioned conclusion, since in these subgroups the established maximum computational time was not reached and the presented gap was 0%. All other solutions belonging to subgroups B3, C2 and C3 are feasible solutions, however they are not optimal, that is, it was not possible to find the solution that would obtain the lowest value for the objective function within the established computational time limit.

As previously stated, not all of the solutions found were optimal, meaning that the difference between the best integer solution and the best node found was greater than 0%. When comparing subgroups C2 and

Table 6.2: Summary of the final results obtained with the exact model

Subgroup	Number of theses	Availability (%)	Time (s)	Gap (%)	Objective function
			-	-	-
A1	33	30	-	-	-
			-	-	-
		Average	-	-	-
			45.09	0	1050.50
A2	33	50	42.30	0	1228.01
			4.56	0	1641.49
		Average	30.65	0	1306.67
			1471.89	0	549.58
A3	33	70	4511.43	0	519.79
			3760.03	0	524.44
		Average	3248.78	0	531.27
			-	-	-
B1	51	30	-	-	-
			-	-	-
		Average	-	-	-
			2080.06	0	2262.59
B2	51	50	912.00	0	2868.95
			574.27	0	2789.72
		Average	1188.78	0	2640.42
			10800.00	35.31	1165.13
B3	51	70	10800.00	36.12	1097.36
			10800.00	36.53	1418.91
		Average	10800.00	35.99	1227.13
			-	-	-
C1	96	30	-	-	-
			-	-	-
		Average	-	-	-
			10800.00	29.98	8647.67
C2	96	50	10800.00	39.45	8286.12
			10800.00	26.91	8462.69
		Average	10800.00	32.11	8465.69
			10800.00	93.39	32809.00
C3	96	70	10800.00	47.59	3982.79
			10800.00	91.96	30250.80
		Average	10800.00	77.65	22347.53

C3, which have the same number of thesis defences to allocate, it is possible to observe an increase in the average gap. This increase is due to a greater availability of professors, which creates more possible combinations for the allocation of thesis defences, leading to an increase in computational effort required to solve the problem. In subgroup C3, it is particularly noteworthy that the average gap was 77.65%, which indicates a significant difference between the presented result and the optimal solution for that problem, even after the model runs for 10800.00 seconds.

Analyzing the results obtained for the objective function of the different instances, it becomes apparent that there is a decrease in the average objective function value within groups A and B. More specifically, the average objective function value of subgroups A2 and B2 is higher than the value of subgroups A3 and B3, respectively. This decrease signifies an improved allocation of thesis defences in accordance with the defined model's characteristics. This improvement is due to the greater availability of members, which results in a larger set of scheduling possibilities. However, this trend is not observed in subgroups C2 and C3, which exhibit average objective function values of 8465.69 and 22347.53, respectively. This is due to the fact that subgroup C3 has a high average gap, which suggests that the solution is far from being optimal and, as a result, not comparable.

We can thus support the fact that the number of defences to allocate and the availability of each member influence the result presented by the model for each instance, given that, keeping the other variables equal, an increase in the number of thesis defences leads to an increase in the objective function value in subgroups A2, B2 and C2, with mean values of 1306.67, 2640.42, 8465.69. On the contrary, an increase in the availability of the members is reflected in a decrease in the objective function, as compared before.

As previously described, the objective function of the model minimizes the number of days each member has to attend thesis defences and the total number of empty slots between thesis defences allocated on the same day for each member. In other words, it is directly related to the variables G_m and H_{md}. The values of these variables were collected and the average value for each subgroup of instances was summarized in Table 6.3. Only subgroups with feasible solutions were considered, i.e., subgroups A2, A3, B2, B3, C2, and C3.

Among the subgroups that presented feasible solutions, the maximum number of days that a member $m \in M$ had to attend thesis defences varied between 3 and 9, depending on the number of defences to be allocated and the availability of the professors. Analyzing the number of empty slots between thesis defences allocated on the same day for each member, it was found that the values ranged from 2 to 4, indicating that the scheduling performed achieved a high degree of compactness, with little discrepancy in the number of slots between the thesis defences of each member for a given day $d \in D$. The best results regarding these two factors were obtained for instances with a smaller number of theses to be allocated and when the professors had greater availability, i.e., for subgroup A3.

Compared to the current manual process used in the department, the model showed significant improvements in optimizing the scheduling of thesis defences. Solutions were obtained within a maximum of

Table 6.3: Summary of the maximum values of the auxiliary variables G_m e H_{md} for the exact model

Subgroup	Average	
	Maximum G_m	Maximum H_{md}
A2	3	3
A3	3	2
B2	4	2
B3	3	3
C2	6	4
C3	9	3

10800.00 seconds, while the manual process can sometimes take more than two weeks. However, the fact that the model does not provide solutions when it does not find a feasible solution can be a drawback when the professors have limited availability. Therefore, an alternative two-phase exact model solution was tested in subsection 6.2.1, with a modification of a hard constraint.

6.2.1 Number of thesis defences Allocated

As previously stated, the mathematical model was unable to generate feasible solutions for subgroups A1, B1, and C1 due to the inability to allocate all thesis defences, as required by the structural constraint (4.1).

To overcome this challenge and obtain feasible solutions for these subgroups, an alternative approach was adopted by modifying the constraint (4.1). Specifically, the revised constraint (6.1) relaxed the obligation to allocate all thesis defences, enabling the model to generate feasible solutions for these subgroups.

$$\sum_{d \in D} \sum_{t \in T} x_{idt} \leq 1, \forall i \in I \tag{6.1}$$

Furthermore, the model was solved in 2 phases, using a two-phase approach, resulting from two different objective functions. In the first phase, the model aims to maximize the allocation of the largest possible number of theses (6.2). In the second phase of model resolution, the minimization of the number of days each professor is allocated to assist the thesis defences and the number of holes between each defence on each day, are now considered again, forming part of the objective function (6.3) and (6.4). In the existing literature it is possible to see the adoption of this type of approach, in the resolution of STP, UCTP and UETP (Alvarez et al., 2002 [11], Liu et al., 2009, [80], Zhang et al., 2010 [147], and Thompson & Dowsland, 1998 [135]).

$$Max \sum_{i \in I} \sum_{d \in D} \sum_{t \in T} x_{idt} \tag{6.2}$$

$$Min \sum_{m \in M} \sum_{q \in Q} (W1_m \cdot GQ_{mq} \cdot q^{EXP}) \tag{6.3}$$

$$Min \sum_{m \in M} \sum_{d \in D} (W2_m \cdot H_{md}) \tag{6.4}$$

Table 6.4 summarizes all the results obtained for instances belonging to subgroups A1, B1 and C1, all with 30% availability for committee members.

Through a two-phased approach it was possible to obtain results for groups of instances that previously did not present feasible solutions. Although it was not possible to allocate all the thesis defences, in subgroup A1, B1 and C1 more than 95%, 80% and 85% of the theses were allocated, on average, obtaining optimal solutions. As a consequence of the low availability of the jury members, only 30%, the number of possibilities for allocation of thesis defences is reduced, so the computational time required for the model to present a solution is low, on average 9.21s for subgroup A1, 12.20s for group B1 and 19.19s for group C1. As expected, the objective functions obtained high values, compared to those obtained for the remaining instances of subgroups A2, A3, B2, C2, B3 and C3, due to the reduced allocation options of each thesis defence, making it difficult to minimize the number of days and holes between every defence, on every day, for every member. The values of the objective functions showed higher values for the subgroups with a greater number of theses to allocate, with average values of 2281.78, 7619.44 and 26244.81 for subgroups A1, B1 and C1, respectively.

As it was possible to conclude, the number of thesis defences to be allocated and the availability of members directly influence the results obtained. The DEG of IST has no influence on the number of thesis defences to be allocated, since this is dependent on the number of students who submit their dissertation by the deadline for each semester. Thus, a parameter that can be changed is the constitution of thesis defences juries. Its formation, carried out prior to the allocation of thesis defences, makes the model have to respect the availability of each member when scheduling the thesis defences, reducing the allocation possibilities, making some solutions infeasible and with objective function values which are very high.

The section 6.3 presents and analyzes the results obtained by the exact mathematical model constructed as an alternative to the model that represents the current situation of allocation of thesis defences by DEG, presented in chapter 5.

6.3 Alternative Exact Model

The exact model built as an alternative to the one presented in Chapter 4 has the ability to allocate different chairpersons, supervisors and additional members to each thesis defence, using the different allocation possibilities, as a way to optimize the presented solution.

In section 6.3 the results obtained through the developed alternative model are presented. The same

Table 6.4: Summary of the final results obtained with the exact model with a two-phased approach

Subgroup	Number of theses	Theses allocated	Time (s)	Gap (%)	Objective function
A1	33	32	9.39	0	629.62
		31	9.16	0	2500.39
		32	9.09	0	3715.32
	Average	32	9.21	0	2281.78
B1	51	41	11.84	0	7424.09
		42	11.84	0	7424.09
		47	12.07	0	8413.76
	Average	43	12.20	0	7619.44
C1	96	84	18.57	0	27080.30
		84	19.93	0	27979.23
		87	19.08	0	23674.90
	Average	85	19.19	0	26244.81

instances generated for the first model were used, disregarding the constitution of the jury of each thesis defence, keeping only the information about the supervisors of each thesis. In this way, it will be possible to compare the results obtained later.

Table 6.5 summarizes the results obtained through the alternative exact model, for the 27 instances generated, grouped by the respective subgroups depending on the number of theses to be allocated and the availability of the members. The values of the computational time, in seconds, the gap, in %, and the value of the objective function, for each instance, are presented, as well as the average of each subgroup, each one consisting of 3 generated instances.

No instance presented infeasible solutions, however the model was only able to present solutions for subgroups A1, A2, B1, B2 and C1, within the stipulated computational time limit. For the remaining subgroups, after running 10800.00 seconds, the model failed to present a feasible solution to the problem, despite managing to calculate best nodes for each instance. The non-presentation of feasible solutions is due to the increase in the number of thesis defences to be allocated, as well as the availability of each member, which increases the number of constraints and variables that constitute the problem, adding to the fact that in the new model, the juries of each defence being defined by the model, and therefore an exponential increase in the complexity of the problem.

Considering the subgroups that obtained feasible solutions for all instances, in all cases the computational time was 10800.00 seconds, established time limit, and for no instance optimal solutions were obtained, only feasible ones. As a result, the gap presented was always greater than 0%. The subgroup with the lowest gap value, on average 40.77%, was A1, the group with the lowest number of thesis defences to be allocated, and 30% availability of the members. The instances of subgroup C1 presented gap values between 94.04% and 97.00%, presenting solutions that were very far from what would be the optimal solution for the problem. The objective functions followed their lowest value for the A2 subgroup, with

Table 6.5: Summary of the final results obtained with the alternative exact model

Subgroup	Number of theses	Availability (%)	Time (s)	Gap (%)	Objective function
A1	33	30	10800.00	40.33	571.79
			10800.00	42.25	589.93
			10800.00	39.72	564.89
		Average	10800.00	40.77	575.54
A2	33	50	10800.00	59.56	345.76
			10800.00	65.05	347.39
			10800.00	66.32	339.87
		Average	10800.00	63.66	344.34
A3	33	70	10800.00	-	-
			10800.00	-	-
			10800.00	-	-
		Average	10800.00	-	-
B1	51	30	10800.00	41.00	1326.97
			10800.00	45.32	1157.38
			10800.00	42.22	1228.97
		Average	10800.00	42.85	1237.77
B2	51	50	10800.00	75.00	1531.00
			10800.00	62.54	1671.45
			10800.00	69.48	1568.79
		Average	10800.00	69.01	1590.41
B3	51	70	10800.00	-	-
			10800.00	-	-
			10800.00	-	-
		Average	10800.00	-	-
C1	96	30	10800.00	97.00	10340.98
			10800.00	94.32	15503.98
			10800.00	96.80	12564.27
		Average	10800.00	96.04	12803.08
C2	96	50	10800.00	-	-
			10800.00	-	-
			10800.00	-	-
		Average	10800.00	-	-
C3	96	70	10800.00	-	-
			10800.00	-	-
			10800.00	-	-
		Average	10800.00	-	-

Table 6.6: Summary of the maximum values of the auxiliary variables G_m e H_{md} for the alternative exact model

Subgroup	Average	
	Maximum G_m	Maximum H_{md}
A1	2	2
A2	2	2
B1	3	3
B2	3	3
C1	5	5

33 thesis defences to be allocated and with availability of 50% of the members. Among the instances with the same percentage of availability of the members, (A1, B1, and C1) and (A2 and B2) the value of the objective function resulted in increasing values for higher numbers of thesis defences to be allocated. Among the subgroups that have the same number of theses to allocate, there are different behaviors regarding the value of the objective functions. The value of the objective function decreases when the availability of members increases, comparing subgroup A1 with A2. However, comparing the values presented by subgroups B1 and B2, there is an increase in the value of the objective function with the increase in the availability percentage of members, which would not be expected. A justification for this behavior is the high gap percentages, representing a significant distance between the presented solution and the optimal solution. It is possible that an increase in the computational time limit would lead to a decrease in the objective function values of the B2 subgroup instances, compared to the values presented by the B1 subgroup instances.

The objective function of each instance are related to the number of days that each member has to attend thesis defences and the number of empty slots between thesis defences on the same day, the objective being the decrease of these two variables denominated by G_m and H_{md}. In order to evaluate these values, the Table 6.6 summarizes the average results obtained, for each subgroup, in which the model was able to present feasible solutions, that is, A1, A2, B1, B2 and C1.

The values of G_m, number of days in which member $m \in M$ belongs to the committee of at least one thesis defence, varies from 2 to 5, with the lowest value in subgroup A1 and A2, both with the lowest number of thesis defences to allocate, and the highest value in subgroup C1, which has 96 thesis defences to schedule. As for, the number of holes between each member's thesis defence, H_{md}, it also varies between 2 and 5, with the lowest value presented in subgroups A1 and A2 and the highest in subgroup C1. The low values of these two variables reflect compact thesis defence allocations for each member, which take into account their preferences, reducing the number of days as much as possible.

6.4 Comparison Between Exact Model and Alternative Exact Model

Different values were obtained through the exact model and the alternative model developed. Section 6.4 compares the results achieved for the different subgroups of instances. This comparison is possible since the instances used to run the different models maintained the same characteristics, with the exception of the formation of different thesis juries. Thus, the influence of the predefinition, or not, of the committees of each thesis defence will be compared. To carry out the comparison, the average values of each subgroup were used, since no instance presented different values from the others belonging to the same group, and therefore the average values are a good representation of the instances of each subgroup.

Table 6.7 represents the mean values obtained through the exact model and alternative exact model, for each subgroup consisting of three different instances. The computational time, gap and objective function values were summarized.

The definition of committees for each thesis defence together with the allocation of defences to one day and timeslot, as performed in the alternative exact model, made it possible to present feasible solutions for subgroups A1, B1 and C1. Through the first model, which received as parameters the constitution of the jury of each thesis defence, thus conditioning the allocation of each defence only to the days and timeslots in which the members belonging to the jury of each one were available, it was unable to present solutions for the subgroups where professors' availability was only 30%, the lowest value of the availability parameter among the different subgroups. However, although the alternative model managed to present solutions for all cases that the first model presented the solution as infeasible, for instances with a higher value of member availability, subgroups A3, B3, C2 and C3, the alternative model was unable to present a solution within the established time limit, due to the high complexity of the problem.

The time that the alternative exact model needed to run the different instances was always equal or greater than the time that the first model needed, due to the high complexity of the problem. The alternative model failed to present any optimal solution, never running less than the established time limit of 10800.00 seconds, while the first model obtained solutions in 30.65 seconds, 1188.78 seconds, 3248.78 seconds and 10800.00 seconds, and for the subgroups A2, A3 and B2 the presented solutions were optimal solutions.

For each subgroup, the first built model always presented gap percentages smaller than the alternative model, with the exception of subgroup C2 with a gap of 77.65%, which means that the solutions presented by the exact model were optimal solutions or more close to being so, than those presented by the alternative exact model, reflected in the presented value of the objective function. Within the subgroups that it was possible to obtain objective function values with both models, A2 and B2, there is always a significant decrease between the value achieved through the exact model and the alternative exact model, with decreases of 74% and 40% of the values obtained through the alternative exact model, for subgroups A2 and B2, respectively. Even with a higher gap, the alternative exact model presents better solutions taking into account professors' preferences, allocating thesis defences in order to obtain more compact timetables for professors. The definition of the committees for each defence, at the same time as the allocation of thesis defences, grants greater flexibility to the process, with a greater number of allocation possibilities, and therefore makes possible a better schedule, compared to that elaborated with the first built model, which exemplifies the department's allocation of thesis defences. In instance group A, subgroups A1, A2 and A3, the lowest value of the objective functions obtained was through the alternative exact model, 344.34. This value was lower than the value obtained by the exact model for instances in which members were less available, as is the case of subgroup A3, which obtained an average value of 531.27. This fact further demonstrates the influence that the formation of committees has on the results obtained.

Table 6.7: Comparison of the results obtained with the exact model and the alternative exact model

Subgroup	Number of theses	Availability (%)	Pre-established committees?	Feasible solution?	Time (s)	Gap (%)	Objective function
A1	33	30	Yes	No	-	-	-
			No	Yes	10800.00	40.77	575.54
A2	33	50	Yes	Yes	30.65	0	1306.67
			No	Yes	10800.00	63.66	344.34
A3	33	70	Yes	Yes	3248.78	0	531.27
			No	Yes	10800.00	-	-
B1	51	30	Yes	No	-	-	-
			No	Yes	10800.00	42.85	1237.77
B2	51	50	Yes	Yes	1188.78	0	2640.42
			No	Yes	10800.00	69.01	1590.41
B3	51	70	Yes	Yes	10800.00	35.99	1227.13
			No	Yes	10800.00	-	-
C1	96	30	Yes	No	-	-	-
			No	Yes	10800.00	96.04	12803.08
C2	96	50	Yes	Yes	10800.00	32.11	8465.49
			No	Yes	10800.00	-	-
C3	96	70	Yes	Yes	10800.00	77.65	22347.53
			No	Yes	10800.00	-	-

Table 6.8: Comparison of variable G_m e H_{md} obtained with the exact model and the alternative exact model

Subgroup	Pre-established committees?	Maximum G_m	Maximum H_{md}
A2	Yes	3	3
	No	2	2
B2	Yes	4	2
	No	3	3

In order to complement the analysis and comparison of the results obtained for each group of instances, the values of the variables G_m and H_{md}, obtained by the first and second exact model developed, were summarized in the table 6.8. It should be noted that all the values presented for each subgroup are the average values of the 3 instances that constitute them. As a second note, it should be mentioned that only variable values were presented for the subgroups in which both models presented feasible solutions.

For both subgroups A2 and B2, the results obtained for the variable G_m were optimized by the alternative model, since the variable had a lower average value, resulting in a smaller number of days in which professors were allocated to assist thesis defences. As for the variable H_{md}, it is not possible to draw conclusions, since the results obtained lead to opposite analyses. These results, as previously mentioned, derive from the high complexity of the problems, resulting in a large computational time needed to find feasible solutions, and therefore a high gap, when the computational time reaches the stipulated limit.

6.5 Conclusion

This chapter presented the results of adaptation of the TDTP of DEG through two exact models, with different characteristics regarding the formation of the committees of each thesis defence. The models were tested through instances generated with characteristics similar to the reality witnessed in the department. The objective of the chapter was to analyze the results obtained through the two models, and to carry out a comparison between them, in order to conclude the best way to carry out the scheduling of thesis defences, manually, through a mathematical model that represents the reality of the department or through a model capable of defining the committees together with the allocation of each thesis defence. The influence of the number of theses to be allocated, the availability of members and the definition of committees before or together with the allocation of thesis defences was analyzed, through variations in the values of the obtained objective functions.

The first model was tested for a time limit of 10800.00 seconds, and managed to obtain optimal solutions for instances with 33 and 51 thesis defences to be allocated, however for subgroups of instances with the parameter availability of members with a value of 30%, the solutions presented were always infeasible, highlighting the impossibility of allocating all thesis defences to instances with low availability values. There was an increase in the value of the objective function with the increasing value of thesis defences to allocate. On the contrary, the increase in availability was reflected in a decrease in the value of the objective functions within the same group, A and B. The group that did not reflected the same behavior was group C, justified by the high value of the gap presented by the subgroup of C3 instances. Through a two-phased approach and changing the hard constraint that made the allocation of all thesis defences mandatory, it was possible to find solutions for instances of subgroups A1, B1 and C1.

The alternative exact model did not find any optimal solution for the instances, always reaching the computational time limit of 10800.00 seconds. It was also not possible to present solutions for instances with availability rates of 70% and for subgroup C2. The formation of juries for each thesis defence simultaneously with the allocation of each thesis defence increases the complexity of the problem, reflecting on the gap values presented, always higher than 40%. There is a direct relation between the increase in the number of thesis defences to be allocated and the increase in the value of the objective function. In turn, there was a decrease in this value when the availability rate increases, within group A. Group B did not show the same behavior, justified by the high values of gaps, reflecting a great distance from the presented solution to the optimal solution.

The comparison of the two models allowed the analysis of the effect of the formation of thesis juries before or simultaneously with the allocation of each defence. This factor influences the feasibility of the solutions as well as the complexity of the problem, and consequently the computational time required to find the optimal solution to the problem. It was verified that a formation of the committees simultaneous to the allocation of the thesis defences makes all the solutions feasible, presenting lower values of the objective functions, that is, the allocation is made in a better way. On the other hand, this variation increases the complexity of the problem, through a greater number of variables and constraints, translating into an increase in the computational time required. There were thus advantages and disadvantages in both models, requiring the decision maker to choose what to prioritize, obtaining better schedules or computational efficiency. Compared to the current way of allocating thesis defences, the mathematical models constructed proved to be more efficient, requiring less time to allocate the thesis defences.

7. Conclusions and Future Work

Scheduling problems constitute a vast and extensively researched area of literature, with studies dating back decades. These problems aim to allocate resources and events to designated timeslots, as indicated by Beligiannis et al., in 2008 [21]. Timetabling problems encompass a range of issues, including Educational Timetabling Problems, Pillay, in 2016 [97].

The literature contains numerous studies that specifically address the STP and UTP, with a focus on UCTP, UETP, and TDTP. The specific constraints considered in each approach are dependent on the particular characteristics of the case study, and as such, the approaches differ between investigations. In addressing these problems, researchers have applied both exact methods, such as Integer Programming Models, Multicommodity Flow Models, Constraint Programming Models, and Mixed Integer Programming Models, as well as non-exact methods.

IST, like many other universities, faces various scheduling problems. These problems are particularly prevalent in departments responsible for master's degrees, such as the scheduling of thesis defences. This thesis specifically focuses on the scheduling problem of DEG, which is responsible for allocating thesis defences for MEGI students. Given the high number of thesis defences that must be allocated each semester, as well as the varying availability of professors, the thesis jury members, this task is complex and time-consuming, particularly during the submission deadline months of May and October. Currently, this task is performed manually by the department's secretary, who collects the availability of each jury member and establishes a common date when all members are available.

To study TDTP, we analyzed the case of DEG and developed two Mixed Integer Linear Programming Models. One of these models reflects the real-world constraints observed in the department, while the other is capable of defining the members of each thesis jury and their allocation. It is worth noting that none of the existing studies in the literature have addressed this issue with the defined models. However, it is possible to find some of the constraints considered in our study in the works of Huynh et al. (2012) [61], Fastré et al. (2017) [51], and Almeida (2021) [8].

One objective function was defined for both models with two components: the first aims to minimize the number of days on which each member of the jury is required to attend thesis defences, while the second aims to minimize the number of empty slots between each thesis defence allocated on the same day, for each professor. These factors are not currently taken into account in the department's manual allocation process.

In order to analyze the effect of the number of thesis defences, professors' availability and the constraints defined on the solution's viability and on the objective function value, instances were generated with different values of defences to allocate and professors' availability.

The analysis of the results obtained from the two models revealed that the complexity of the problem increases with an increase in the number of thesis defences to be allocated and the availability of professors. Consequently, the computational time required to find a feasible solution also increases, which may render finding optimal solutions impossible. It was observed that a higher percentage of availability of professors led to better results in the model, resulting in more condensed schedules for the professors. Conversely, an increase in the number of thesis defences led to an adverse effect.

The requirement to allocate all thesis defences can lead to the infeasibility of the model. However, by adding a new objective function that maximizes the allocation of thesis defences, changing the hard constraint constructed, it was possible to obtain optimal solutions within an acceptable computational time using the first model.

The comparison of results obtained with the two models revealed both advantages and disadvantages in simultaneously defining the committees of each thesis defence and their allocation. This modification added an extra level of complexity to the problem, resulting in an exponential increase in computational time when compared to the current method employed by the department. The model was unable to provide solutions within a reasonable computational time when dealing with a large number of defences to be allocated. However, the increased flexibility in the allocation of professors and, subsequently, of the thesis defences, enabled the allocation of all defences, regardless of the availability of professors, and provided more favorable schedules for them.

The proposed models provide an automated solution for scheduling thesis defences, offering improvements in terms of both time required to find a feasible solution and the schedules generated, which consider the preferences of the professors.

Therefore, depending on the number of thesis defences to be allocated, either the second model or the first model can be utilized by the department. The second model is suitable if the number of thesis defences is low, whereas the first model is recommended if committees are established before their allocation. In both cases, the workload of the secretary would be reduced, and the time required to find a solution would decrease, resulting in better scheduling for the professors.

As a line of future research, It would be interesting to analyze the results obtained by the two models on real data from the department, allowing the allocation of MEGI thesis defences to a semester and the construction of compact timetables for DEG professors. In this way, it would be possible to analyze the effectiveness of the problem in real-size instances, allowing feedback from the members involved on the efficiency of the models and the defined schedules. The decision maker could propose improvements to be made in the model considering that other types of constraints would be more practical to place in its definition.

In addition, due to the lengthy computational time required by the models to provide solutions for larger instances with higher availability levels, it would be beneficial to examine the outcomes achieved for the same types of instances using non-exact models. The literature reveals promising findings in utilizing matheuristics for scheduling problems, particularly in STP and UTP. Employing a combination of mathematical programming techniques, heuristics, and meta-heuristics can be an effective methodology, which involves applying heuristics to the Mixed Integer Linear Programming models defined, to achieve more efficient models for instances with higher numbers.

Bibliography

[1] S. Abdullah, E. K Burke, and B. McCollum. “Using a randomised iterative improvement algorithm with composite neighbourhood structures for the university course timetabling problem”. In: *Metaheuristics*. Springer, 2007, pp. 153–169.

[2] S. Abdullah, E. K Burke, and B. Mccollum. “An investigation of variable neighbourhood search for university course timetabling”. In: *The 2th Multidisciplinary Conference on Scheduling: Theory and Applications, NY, USA*. Citeseer. 2005, pp. 413–427.

[3] D. Abramson. “Constructing school timetables using simulated annealing: sequential and parallel algorithms”. In: *Management science* 37.1 (1991), pp. 98–113.

[4] D. Abramson and J. Abela. *A parallel genetic algorithm for solving the school timetabling problem*. 1991.

[5] L. Ahmed, E. Özcan, and A. Kheiri. “Solving high school timetabling problems worldwide using selection hyper-heuristics”. In: *Expert Systems with Applications* 42.13 (2015), pp. 5463–5471.

[6] C. Aladag, G. Hocaoglu, and M. Basaran. “The effect of neighborhood structures on tabu search algorithm in solving course timetabling problem”. In: *Expert systems with applications* 36.10 (2009), pp. 12349–12356.

[7] P. Alefragis et al. “A multiple metaheuristic variable neighborhood search framework for the Uncapacitated Examination Timetabling Problem”. In: *Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT*. Vol. 1. 2021, pp. 159–171.

[8] J. Almeida. “An integer programming model for scheduling master’s thesis defences.” In: (2021).

[9] O. Alsmadi et al. “A novel genetic algorithm technique for solving university course timetabling problems”. In: *International Workshop on Systems, Signal Processing and their Applications, WOSSPA*. IEEE. 2011, pp. 195–198.

[10] R. Alvarez-Valdes, E. Crespo, and J. Tamarit. “Design and implementation of a course scheduling system using tabu search”. In: *European Journal of Operational Research* 137.3 (2002), pp. 512–523.

[11] R. Alvarez-Valdés, F. Parreño, and J. Tamarit. “A tabu search algorithm for assigning teachers to courses”. In: *Top* 10.2 (2002), pp. 239–259.

[12] M. Alzaqebah and S. Abdullah. “Hybrid bee colony optimization for examination timetabling problems”. In: *Computers & Operations Research* 54 (2015), pp. 142–154.

[13] Khairul Anwar et al. “Harmony search-based hyper-heuristic for examination timetabling”. In: *2013 IEEE 9th international colloquium on signal processing and its applications*. IEEE. 2013, pp. 176–181.

[14] Pasquale Avella and Igor Vasil’Ev. “A computational study of a cutting plane algorithm for university course timetabling”. In: *Journal of Scheduling* 8.6 (2005), pp. 497–514.

[15] E. Aycan and T. Ayav. “Solving the course scheduling problem using simulated annealing”. In: *2009 IEEE International Advance Computing Conference*. IEEE. 2009, pp. 462–466.

[16] Z. Azimi. “Comparison of metaheuristic algorithms for examination timetabling problem”. In: *Journal of Applied Mathematics and computing* 16.1 (2004), pp. 337–354.

[17] H. Babaei, J. Karimpour, and A. Hadidi. “A survey of approaches for university course timetabling problem”. In: *Computers & Industrial Engineering* 86 (2015), pp. 43–59.

[18] A. Bajeh and K. Abolarinwa. “Optimization: A comparative study of genetic and tabu search algorithms”. In: *International Journal of Computer Applications* 31.5 (2011), pp. 43–48.

[19] M. Bakir and C. Aksop. “A 0-1 integer programming approach to a university timetabling problem”. In: *Hacettepe Journal of Mathematics and Statistics* 37.1 (2008), pp. 41–55.

[20] M. Battistutta et al. “Modelling and solving the thesis defense timetabling problem”. In: *Journal of the Operational Research Society* 70.7 (2019), pp. 1039–1050.

[21] G. N Beligiannis et al. “Applying evolutionary computation to the school timetabling problem: The Greek case”. In: *Computers & Operations Research* 35.4 (2008), pp. 1265–1280.

[22] R. Bellio et al. “Two-stage multi-neighborhood simulated annealing for uncapacitated examination timetabling”. In: *Computers & Operations Research* 132 (2021), p. 105300.

[23] T. Birbas, S. Daskalaki, and E. Housos. “School timetabling for quality student and teacher schedules”. In: *Journal of Scheduling* 12.2 (2009), pp. 177–197.

[24] T. Birbas, S. Daskalaki, and E. Housos. “Timetabling for Greek high schools”. In: *Journal of the Operational Research Society* 48.12 (1997), pp. 1191–1200.

[25] N. Boland et al. “New integer linear programming approaches for course timetabling”. In: *Computers & Operations Research* 35.7 (2008), pp. 2209–2233.

[26] E. Burke et al. “A branch-and-cut procedure for the Udine course timetabling problem”. In: *Annals of Operations Research* 194.1 (2012), pp. 71–87.

[27] E. Burke et al. “Hybrid variable neighbourhood approaches to university exam timetabling”. In: *European Journal of Operational Research* 206.1 (2010), pp. 46–53.

[28] V. Cacchiani and T. Paolo. “Nominal and robust train timetabling problems”. In: *European Journal of Operational Research* 219.3 (2012), pp. 727–737.

[29] M. Carrasco and M. Pato. “A multiobjective genetic algorithm for the class/teacher timetabling problem”. In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer. 2000, pp. 3–17.

[30] M. Carter and G. Laporte. “Recent developments in practical examination timetabling”. In: *International conference on the practice and theory of automated timetabling*. Springer. 1995, pp. 1–21.

[31] M. Carter and C. Tovey. “When is the classroom assignment problem hard?” In: *Operations Research* 40.1-supplement-1 (1992), S28–S39.

[32] A. Cataldo et al. “An integer programming approach to curriculum-based examination timetabling”. In: *Annals of Operations Research* 258.2 (2017), pp. 369–393.

[33] S. Ceschia, L. Di Gaspero, and A. Schaerf. “Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem”. In: *Computers & Operations Research* 39.7 (2012), pp. 1615–1624.

[34] M. Chen et al. “A Tabu search algorithm with controlled randomization for constructing feasible university course timetables”. In: *Computers & Operations Research* 123 (2020), p. 105007.

[35] G. Christopher and A. Wicaksana. “Particle swarm optimization for solving thesis defense timetabling problem.” In: *Telkomnika* 19.3 (2021).

[36] A. Colorni, M. Dorigo, and V. Maniezzo. “A genetic algorithm to solve the timetable problem”. In: *Politecnico di Milano, Milan, Italy TR* (1992), pp. 90–060.

[37] A. Colorni, M. Dorigo, and V. Maniezzo. “Genetic algorithms and highly constrained problems: The time-table case”. In: *International Conference on Parallel Problem Solving from Nature*. 1990, pp. 55–59.

[38] A. Colorni, M. Dorigo, and V. Maniezzo. “Metaheuristics for high school timetabling”. In: *Computational optimization and applications* 9.3 (1998), pp. 275–298.

[39] J. Cordeau, B. Jaumard, and R. Morales. “Efficient Timetabling Solution with Tabu Search”. In: (2003).

[40] G. Da Fonseca et al. “GOAL solver: a hybrid local search based solver for high school timetabling”. In: *Annals of Operations Research* 239.1 (2016), pp. 77–97.

[41] S. Daskalaki, T. Birbas, and E. Housos. “An integer programming formulation for a case study in university timetabling”. In: *European journal of operational research* 153.1 (2004), pp. 117–135.

[42] E. Demirović and N. Musliu. “MaxSAT-based large neighborhood search for high school timetabling”. In: *Computers & Operations Research* 78 (2017), pp. 172–180.

[43] E. Demirović and P. Stuckey. “Constraint programming for high school timetabling: a scheduling-based model with hot starts”. In: *International conference on the integration of constraint programming, artificial intelligence, and operations research*. Springer. 2018, pp. 135–152.

[44] L. Di Gaspero. “Recolour, shake and kick: A recipe for the examination timetabling problem”. In: *Proc. of the 4th Int. Conf. on the Practice and Theory of Automated Timetabling*. 2002, pp. 404–407.

[45] L. Di Gaspero and A. Schaerf. “Tabu search techniques for examination timetabling”. In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer. 2000, pp. 104–117.

[46] J. Domrös and J. Homberger. “An evolutionary algorithm for high school timetabling”. In: *Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012)*. Citeseer. 2012, pp. 485–488.

[47] Á. Dorneles, O. de Araújo, and L. Buriol. "A column generation approach to high school timetabling modeled as a multicommodity flow problem". In: *European Journal of Operational Research* 256.3 (2017), pp. 685–695.

[48] Á. Dorneles, O. de Araújo, and L. Buriol. "A fix-and-optimize heuristic for the high school timetabling problem". In: *Computers & Operations Research* 52 (2014), pp. 29–38.

[49] P. Dung et al. "A Java library for Constraint-Based Local Search: Application to the master thesis defense timetabling problem". In: *Proceedings of the Sixth International Symposium on Information and Communication Technology*. 2015, pp. 67–74.

[50] M. Eley. "Ant algorithms for the exam timetabling problem". In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer. 2006, pp. 364–382.

[51] L. Fastré, P. Schaus, and F. Glineur. "Automatic scheduling for master thesis defenses". In: (2017).

[52] C. Fernandes et al. "High school weekly timetabling by evolutionary algorithms". In: *Proceedings of the 1999 ACM symposium on Applied computing*. 1999, pp. 344–350.

[53] G. Filho and L. Lorena. "A constructive evolutionary approach to school timetabling". In: *Applications of evolutionary computing* (2001), pp. 130–139.

[54] G. Fonseca and H. Santos. "Variable neighborhood search based algorithms for high school timetabling". In: *Computers & Operations Research* 52 (2014), pp. 203–208.

[55] G. Fonseca, H. Santos, and E. Carrano. "Integrating matheuristics and metaheuristics for timetabling". In: *Computers & Operations Research* 74 (2016), pp. 108–117.

[56] G. Fonseca et al. "Integer programming techniques for educational timetabling". In: *European Journal of Operational Research* 262.1 (2017), pp. 28–39.

[57] S. Goh et al. "An effective hybrid local search approach for the post enrolment course timetabling problem". In: *Opsearch* 57.4 (2020), pp. 1131–1163.

[58] C. Gotlieb. "The Construction of Class-Teacher Time-Tables". In: *IFIP Congress*. 1962.

[59] P. Hinchliffe. "Movement Problems in Comprehensive Schools". In: *Journal of the Operational Research Society* 24.2 (1973), pp. 235–240.

[60] F. Huédé et al. "MCS—A new algorithm for multicriteria optimisation in constraint programming". In: *Annals of Operations Research* 147.1 (2006), pp. 143–174.

[61] T. Huynh, Q. Pham, and D. Pham. "Genetic algorithm for solving the master thesis timetabling problem with multiple objectives". In: *2012 Conference on Technologies and Applications of Artificial Intelligence*. IEEE. 2012, pp. 74–79.

[62] S. Innet. "A noval approach of genetic algorithm for solving examination timetabling problems: A case study of Thai Universities". In: *2013 13th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE. 2013, pp. 233–237.

[63] N. Ismayilova, M. Sağir, and R. Gasimov. "A multiobjective faculty–course–time slot assignment problem with preferences". In: *Mathematical and Computer Modelling* 46.7-8 (2007), pp. 1017–1029.

[64] F. Jacobsen, A. Bortfeldt, and H. Gehring. “Timetabling at German secondary schools: tabu search versus constraint programming”. In: *in Proceedings 6th international conference on the practice and theory of automated timetabling, PATAT2006*. Citeseer. 2006, pp. 439–442.

[65] A. Jamili et al. “Developing a comprehensive and multi-objective mathematical model for university course timetabling problem: a real case study”. In: *Conference: proceedings of the international conference on industrial engineering and operations management, Paris, France*. Vol. 130. 2018.

[66] S. Jat and S. Yang. “A memetic algorithm for the university course timetabling problem”. In: *2008 20th IEEE International Conference on Tools with Artificial Intelligence*. Vol. 1. IEEE. 2008, pp. 427–433.

[67] S. Jha. “Exam timetabling problem using genetic algorithm”. In: *International Journal of Research in Engineering and Technology* 3.5 (2014), pp. 649–654.

[68] L. Jourdan, M. Basseur, and E. Talbi. “Hybridizing exact methods and metaheuristics: A taxonomy”. In: *European Journal of Operational Research* 199.3 (2009), pp. 620–629.

[69] M. Kalender et al. “A greedy gradient-simulated annealing hyper-heuristic for a curriculum-based course timetabling problem”. In: *2012 12th UK Workshop on Computational Intelligence (UKCI)*. IEEE. 2012, pp. 1–8.

[70] A. Kheiri and E. Keedwell. “A hidden markov model approach to the problem of heuristic selection in hyper-heuristics with a case study in high school timetabling problems”. In: *Evolutionary computation* 25.3 (2017), pp. 473–501.

[71] A. Kheiri, E. Özcan, and A. Parkes. “A stochastic local search algorithm with adaptive acceptance for high-school timetabling”. In: *Annals of Operations Research* 239.1 (2016), pp. 135–151.

[72] P. Khonggamnerd and S. Innet. “On improvement of effectiveness in automatic university timetabling arrangement with applied genetic algorithm”. In: *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*. 2009, pp. 1266–1270.

[73] B. Kochaniková and H. Rudová. “Student scheduling for bachelor state examinations”. In: *Proceedings of the 6th Multidisciplinary International Scheduling Conference–MISTA*. Citeseer. 2013, pp. 762–766.

[74] M. Kohshori and M. Abadeh. “Hybrid genetic algorithms for university course timetabling”. In: *International Journal of Computer Science Issues (IJCSI)* 9.2 (2012), p. 446.

[75] S. Kristiansen, M. Sørensen, and T. Stidsen. “Integer programming for the generalized high school timetabling problem”. In: *Journal of Scheduling* 18.4 (2015), pp. 377–392.

[76] S. Kristiansen and TR. Stidsen. “A comprehensive study of educational timetabling, a survey”. In: *Department of Management Engineering, Technical University of Denmark.(DTU Management Engineering Report* 8 (2013).

[77] N. Lawrie. “An integer linear programming model of a school timetabling problem”. In: *The Computer Journal* 12.4 (1969), pp. 307–316.

[78] N. Leite, F. Melıcio, and A. Rosa. “A fast simulated annealing algorithm for the examination timetabling problem”. In: *Expert Systems with Applications* 122 (2019), pp. 137–151.

[79] M. Lindahl, M. Sørensen, and T. Stidsen. “A fix-and-optimize matheuristic for university timetabling”. In: *Journal of Heuristics* 24.4 (2018), pp. 645–665.

[80] Y. Liu, D. Zhang, and S. Leung. “A simulated annealing algorithm with a new neighborhood structure for the timetabling problem”. In: *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*. 2009, pp. 381–386.

[81] M. Marte. “Models and algorithms for school timetabling: A constraint programming approach”. PhD thesis. München, Univ., Diss., 2002, 2002.

[82] B. McCollum et al. “A new model for automated examination timetabling”. In: *Annals of Operations Research* 194.1 (2012), pp. 291–315.

[83] A. Meisels and A. Schaerf. “Modelling and solving employee timetabling problems”. In: *Annals of Mathematics and Artificial Intelligence* 39.1 (2003), pp. 41–59.

[84] F. Melıcio et al. “THOR: a tool for school timetabling”. In: *Proceedings of the 6th International Conference on the Practice and Teaching of Automated Timetabling (PATAT 2006)*. 2006, pp. 532–535.

[85] L. Merlot et al. “A hybrid algorithm for the examination timetabling problem”. In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer. 2002, pp. 207–231.

[86] R. Mikkelsen and D. Holm. “A parallelized matheuristic for the International Timetabling Competition 2019”. In: *Journal of Scheduling* (2022), pp. 1–24.

[87] K. Minh et al. “Using tabu search for solving a high school timetabling problem”. In: *Advances in intelligent information and database systems*. 2010, pp. 305–313.

[88] S. MirHassani. “Improving paper spread in examination timetables using integer programming”. In: *Applied Mathematics and Computation* 179.2 (2006), pp. 702–706.

[89] K. Nguyen et al. “Variable neighborhood search for a real-world curriculum-based university timetabling problem”. In: *2011 Third International Conference on Knowledge and Systems Engineering*. IEEE. 2011, pp. 157–162.

[90] C. Nothegger et al. “Solving the post enrolment course timetabling problem by ant colony optimization”. In: *Annals of Operations Research* 194.1 (2012), pp. 325–339.

[91] K. Nurmi and J. Kyngas. “A conversion scheme for turning a curriculum-based timetabling problem into a school timetabling problem”. In: *7th International Conference on the Practice and Theory of Automated Timetabling, PATAT 2008* (2008), pp. 1–15.

[92] K. Nurmi and J. Kyngas. “A framework for school timetabling problem”. In: *Proceedings of the 3rd multidisciplinary international scheduling conference: theory and applications, Paris*. 2007, pp. 386–393.

[93] M. Ozdemir and R. Gasimov. “The analytic hierarchy process and multiobjective 0–1 faculty course assignment”. In: *European Journal of Operational Research* 157.2 (2004), pp. 398–408.

[94] T. Pais and E. Burke. “Choquet integral for combining heuristic values for exam timetabling problem”. In: *PATAT 2010* (2010), p. 305.

[95] K. Papoutsis, C. Valouxis, and E. Housos. “A column generation approach for the timetabling problem of Greek high schools”. In: *Journal of the Operational Research Society* 54.3 (2003), pp. 230–238.

[96] A. Phillips et al. “Integer programming methods for large-scale practical classroom assignment problems”. In: *Computers & Operations Research* 53 (2015), pp. 42–53.

[97] N. Pillay. “A review of hyper-heuristics for educational timetabling”. In: *Annals of Operations Research* 239.1 (2016), pp. 3–38.

[98] N. Pillay. “A survey of school timetabling research”. In: *Annals of Operations Research* 218.1 (2014), pp. 261–293.

[99] N. Pillay and W. Banzhaf. “A study of heuristic combinations for hyper-heuristic systems for the uncapacitated examination timetabling problem”. In: *European Journal of Operational Research* 197.2 (2009), pp. 482–491.

[100] N. Pillay and W. Banzhaf. “An informed genetic algorithm for the examination timetabling problem”. In: *Applied Soft Computing* 10.2 (2010), pp. 457–467.

[101] G. Post et al. “An XML format for benchmarks in high school timetabling”. In: *Annals of Operations Research* 194.1 (2012), pp. 385–397.

[102] R. Qu and E. Burke. “Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems”. In: *Journal of the Operational Research Society* 60.9 (2009), pp. 1273–1285.

[103] R. Qu, E. Burke, and B McCollum. “Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems”. In: *European Journal of Operational Research* 198.2 (2009), pp. 392–404.

[104] A. Qualizza and P. Serafini. “A column generation scheme for faculty timetabling”. In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer. 2004, pp. 161–173.

[105] R. Raghavjee and N. Pillay. “An application of genetic algorithms to the school timetabling problem”. In: *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*. 2008, pp. 193–199.

[106] S. Rahman et al. “Adaptive linear combination of heuristic orderings in constructing examination timetables”. In: *European Journal of Operational Research* 232.2 (2014), pp. 287–297.

[107] P. Rattadilok. “An investigation and extension of a hyper-heuristic framework”. In: *Informatica* 34.4 (2010).

[108] L. Reis and E. Oliveira. “A constraint logic programming approach to examination scheduling”. In: *Proc. Xth Irish Conference on Artificial Intelligence & Cognitive Science*. 1999.

[109] A. Rezaeipanah, S. Matoori, and G. Ahmadi. “A hybrid algorithm for the university course timetabling problem using the improved parallel genetic algorithm and local search”. In: *Applied Intelligence* 51.1 (2021), pp. 467–492.

[110] S. Ribić and S. Konjicija. “A two phase integer linear programming approach to solving the school timetable problem”. In: *Proceedings of the ITI 2010, 32nd International Conference on Information Technology Interfaces*. IEEE. 2010, pp. 651–656.

[111] P. Ross and J. Marín-Blazquez. “Constructive hyper-heuristics in class timetabling”. In: *2005 IEEE Congress on Evolutionary Computation*. Vol. 2. IEEE. 2005, pp. 1493–1500.

[112] O. Rossi-Doria and B. Paechter. “A memetic algorithm for university course timetabling”. In: *Combinatorial optimisation* (2004), p. 56.

[113] O. Rossi-Doria and B. Paechter. “An hyperheuristic approach to course timetabling problem using an evolutionary algorithm”. In: *a a* 2 (2012), p. 1.

[114] N. Sabar et al. “A graph coloring constructive hyper-heuristic for examination timetabling problems”. In: *Applied Intelligence* 37.1 (2012), pp. 1–11.

[115] H. Santos, L. Ochi, and M. Souza. “An efficient tabu search heuristic for the school timetabling problem”. In: *International Workshop on Experimental and Efficient Algorithms*. Springer. 2004, pp. 468–481.

[116] Ha. Santos et al. “Strong bounds with cut and column generation for class-teacher timetabling”. In: *Annals of Operations Research* 194.1 (2012), pp. 399–412.

[117] L. Saviniec, A. Constantino, and W. Romao. “Vns based algorithms to the high school timetabling problem”. In: *Anais do XLV simpósio brasileiro de pesquisa operacional* (2013), pp. 845–856.

[118] L. Saviniec, M. Santos, and A. Costa. “Parallel local search algorithms for high school timetabling problems”. In: *European Journal of Operational Research* 265.1 (2018), pp. 81–98.

[119] A. Schaerf. “A survey of automated timetabling”. In: *Artificial intelligence review* 13.2 (1999), pp. 87–127.

[120] K. Schimmelpfeng and s. Helber. “Application of a real-world university-course timetabling model solved by integer programming”. In: *Or Spectrum* 29.4 (2007), pp. 783–803.

[121] E. Sin and N. Kham. “Hyper heuristic based on great deluge and its variants for exam timetabling problem”. In: *arXiv preprint arXiv:1202.1891* (2012).

[122] V. I Skoullis, I. Tassopoulos, and G. Beligiannis. “Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm”. In: *Applied Soft Computing* 52 (2017), pp. 277–289.

[123] K. Socha, M. Sampels, and M. Manfrin. “Ant algorithms for the university course timetabling problem with regard to the state-of-the-art”. In: *Workshops on Applications of Evolutionary Computation*. Springer. 2003, pp. 334–345.

[124] A. Soghier and R. Qu. “Adaptive selection of heuristics for assigning time slots and rooms in exam timetables”. In: *Applied Intelligence* 39.2 (2013), pp. 438–450.

[125] M. Sørensen and F. Dahms. “A two-stage decomposition of high school timetabling applied to cases in Denmark”. In: *Computers & Operations Research* 43 (2014), pp. 36–49.

[126] M. Sørensen, S. Kristiansen, and T. Stidsen. “International timetabling competition 2011: An adaptive large neighborhood search algorithm”. In: *Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012)*. 2012, p. 489.

[127] M. Sørensen and T. Stidsen. “High School Timetabling: Modeling and solving a large number of cases in Denmark”. In: *Proceedings of the ninth international conference on the practice and theory of automated timetabling (PATAT 2012)*. 2012, pp. 359–364.

[128] M. Sørensen and T. Stidsen. “Hybridizing integer programming and metaheuristics for solving high school timetabling”. In: *Proceedings of the 10th international conference of the practice and theory of automated timetabling*. 2014, pp. 557–560.

[129] J. Soria-Alcaraz et al. “Effective learning hyper-heuristics for the course timetabling problem”. In: *European Journal of Operational Research* 238.1 (2014), pp. 77–86.

[130] J. Soria-Alcaraz et al. “Iterated local search using an add and delete hyper-heuristic for university course timetabling”. In: *Applied Soft Computing* 40 (2016), pp. 581–593.

[131] S. Sutar and R. Bichkar. “Genetic algorithms based timetabling using knowledge augmented operators”. In: *International Journal of Computer Science and Information Security* 14.11 (2016), p. 570.

[132] J.S. Tan et al. “A survey of the state-of-the-art of optimisation methodologies in school timetabling problems”. In: *Expert Systems with Applications* 165 (2021), p. 113943.

[133] I. Tassopoulos, C. Iliopoulou, and G. Beligiannis. “Solving the Greek school timetabling problem by a mixed integer programming model”. In: *Journal of the Operational Research Society* 71.1 (2020), pp. 117–132.

[134] T. Thepphakorn, P. Pongcharoen, and C. Hicks. “An ant colony based timetabling tool”. In: *International Journal of Production Economics* 149 (2014), pp. 131–144.

[135] J. Thompson and K. Dowsland. “A robust simulated annealing based examination timetabling system”. In: *Computers & Operations Research* 25.7-8 (1998), pp. 637–648.

[136] J. Thompson and K. Dowsland. “Variants of simulated annealing for the examination timetabling problem”. In: *Annals of Operations research* 63.1 (1996), pp. 105–128.

[137] M. Trick. “A schedule-then-break approach to sports timetabling”. In: *International conference on the practice and theory of automated timetabling*. Springer. 2000, pp. 242–253.

[138] M. Tuga, R. Berretta, and A. Mendes. “A hybrid simulated annealing with kempe chain neighborhood for the university timetabling problem”. In: *6th IEEE/ACIS international conference on computer and information science (ICIS 2007)*. IEEE. 2007, pp. 400–405.

[139] C. Valouxis and E. Housos. “Constraint programming approach for school timetabling”. In: *Computers & Operations Research* 30.10 (2003), pp. 1555–1572.

[140] G. White and B. Xie. “Examination timetables and tabu search with longer-term memory”. In: *International Conference on the Practice and Theory of Automated Timetabling*. Springer. 2000, pp. 85–103.

[141] P. Wilke and J. Ostler. “Solving the school timetabling problem using tabu search, simulated annealing, genetic and branch & bound algorithms”. In: *the proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008), Montreal*. Citeseer. 2008.

[142] T. Wong, P. Côté, and P. Gely. “Final exam timetabling: a practical approach”. In: *IEEE CCECE2002. Canadian Conference on Electrical and Computer Engineering. Conference Proceedings (Cat. No. 02CH37373)*. Vol. 2. IEEE. 2002, pp. 726–731.

[143] J. Wood and D. Whitaker. “Student centred school timetabling”. In: *Journal of the Operational Research Society* 49.11 (1998), pp. 1146–1152.

[144] A. Wren. “Scheduling, timetabling and rostering—a special relationship?” In: *International conference on the practice and theory of automated timetabling*. Springer. 1995, pp. 46–75.

[145] S. Al-Yakoob and H. Sherali. “Mathematical models and algorithms for a high school timetabling problem”. In: *Computers & Operations Research* 61 (2015), pp. 56–68.

[146] S. Al-Yakoob, H. Sherali, and M. Al-Jazzaf. “A mixed-integer mathematical modeling approach to exam timetabling”. In: *Computational Management Science* 7.1 (2010), pp. 19–46.

[147] D. Zhang et al. “A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems”. In: *European journal of operational research* 203.3 (2010), pp. 550–558.

A. Appendix A

The present thesis provides an extensive overview of various studies conducted on the themes of STP, UCTP, UETP, and TDTP. Detailed information about these studies can be found in Tables A.1, A.2, A.3, and A.4, respectively. These tables comprise a comprehensive collection of information gathered from diverse sources, offering valuable insights into the various facets of the aforementioned themes.

Table A.1: Summary of STP

Paper	Type of Problem	Solution Approach	Details	Application
Lawrie, 1969	Schedule large units of departments, year groups and layouts	Integer Programming	First feasible solution found	
Papoutsis et al., 2003		Integer Programming	Column Generation	Real world data, Greece
Boland et al., 2008	Choose the students for each class and assign them to timeslots	Integer Programming	Two models were presented	Real world data, Ausatralia
Wilke & Ostler, 2008		Integer Programming	Branch and Bound Algorithm	Real world data, Germamy
Birbas et al., 2009	Assign classes, professors and rooms to available timeslots	Integer Programming		Real world data, Greece
Ribic & Konjicija, 2010	Assign classes to timeslots	Integer Programming	Two-stage approach	
Sorensen & Dahms, 2014	Assign event to timeslots and rooms	Integer Programming	Two-stage approach	Lectio instances

Santos et al., 2012	Assign professors and classes to timeslots	Mixed Integer Linear Programming	Cuts and Column Generation	Real world data, Brazil
Kristiansen et al., 2015	Assign resources and timeslots to events	Mixed Integer Linear Programming		ITC instances
Al-Yakoob & Sherali, 2015	Assign teachers to timeslots and classes	Mixed Integer Linear Programming	Two-stage approach and Column Generation	Real world data, Kuwait
Fonseca et al., 2017	Assign resources and timeslots to events	Mixed Integer Linear Programming		ITC instances
Dorneles et al., 2017	Assign professors and classes to timeslots	Mixed Integer Linear Programming	Column Generation	ITC instances
Tassopoulose et al., 2020	Assign resources and timeslots to events	Mixed Integer Linear Programming		Real world data, Greece
Marte, 2002	Assign classes and professors to timeslots	Constraint Programming		Real world data, Germany
Valouxis & Housos, 2003	Assign classes to timeslots	Constraint Programming	Combinatorial Optimization and Local Search	Real world data, Greece

Demirovic & Stuckey, 2018	Assign resources and timeslots to events	Constraint Programming	Hot Starts used	ITC instances
Colorni et al., 1990	Assign timeslots to events	Genetic Algorithm	Genetic Operators	Real world data, Italy
Abramson & Abela, 1991	Assign resources and timeslots to events	Genetic Algorithm		
Fernandes et al., 1999		Genetic Algorithm	Chromosome representation and Repair Function used	Real world data, Portugal
Filho & Lorena, 2001	Assign resources and timeslots to events	Genetic Algorithm	Constructive Genetic Algorithm	Real world data, Brazil
Nurmi & Kyngas, 2007	Assign timeslots to events	Genetic Algorithm		ITC instances
Raghavjee & Pillay, 2008	Assign resources and timeslots to events	Genetic Algorithm	Crossover and mutation operators	Random instances
Domros & Homberger, 2012		Genetic Algorithm	Randomised method	ITC instances

Sutar & Bichkar, 2016	Assign resources and timeslots to events	Genetic Algorithm	Operators with augmented knowledge	
Sorensen et al., 2012		Adaptive Large Neighborhood	Remove, adaptive and accept strategy	ITC instances
Sorensen & Stidsen, 2012	Assign rooms and timeslots to events	Adaptive Large Neighborhood	Insertion and remove operators	Real world data, Denmark
Saviniec et al., 2013	Assign events to timeslots	Variable Neighborhood Search	Torque and matching operators	Real world data, Brazil
Fonseca & Santos, 2014	Assign resources and timeslots to events	Variable Neighborhood Search	Reduced, Skewed and Sequencial variable neighborhood search	ITC instance
Saviniec et al., 2018	Assign timeslots to events	Parallel Local Search	Central and Diversification-intensification memory based	Real world data, Brazil
Alvarez et al., 2002	Assign professors and timeslots to events	Tabu Search	two-phase approach	Real world data, Spain

Santos et al., 2004		Tabu Search	Two memory diversification scheme	Real world data, Brazil
Jacobsen et al., 2006	Assign timeslots to events	Tabu Search	Period-Neighborhood e Room-Neighborhood	Real world data, Germany
Minh et al., 2010	Assign timeslots to events	Tabu Search	Single, swap and block-changing moves	Real world data, Vietnam
Abramson, 1991	Assign timeslots to events	Simulated Annealing	Monte-Carlo Optimization	Real world data, Australia
Melicio et al., 2006	Assign timeslots to events	Simulated Annealing	Fast Simulated Annealing and Heuristic Construction	Real world data, Portugal
Liu et al., 2009	Assign resources and timeslots to events	Simulated Annealing	two-phase approach	
Zhang et al., 2010	Assign resources and timeslots to events	Simulated Annealing	two-phase approach	
Dorneles et al., 2014	Assign timeslots to events	Matheuristic	Fix-and-Optimize heuristic with VND	

Sorensen & Stidsen, 2014		Matheuristic		ITC instances
Fonseca et al., 2016	Assign resources and timeslots to events	Matheuristic	Variable Neighborhood Search algorithm with Mathematical Programming-based Neighborhoods	
Ahmed et al., 2015	Assign resources and timeslots to events	Hyper-heuristc	Selection Hyper-heuristcs	
Kheiri et al., 2016	Assign resources and timeslots to events	Hyper-heuristc	Multi-stage selection hyper-heuristic	ITC instances
Kheiri & Keedwell, 2017	Assign resources and timeslots to events	Hyper-heuristc	Sequence-based Selection Hyper-heuristic	ITC instances
Da Fonseca et al., 2016	Assign resources and timeslots to events	Hybrid Approach	Simulated Annealing and Iterated Local Search	ITC instances
Demirovic & Musliu, 2017	Assign resources and timeslots to events	Hybrid Approach	Local search with maxSAT-based large neighborhood search	ITC instances

Skoullis et al., 2017[122]	Assign resources and timeslots to events	Hybrid Approach	Hybrid Cat Swarm Optimization	Real world data, Greece

Table A.2: Summary of UCTP

Paper	Type of Problem	Solution Approach	Details	Application
Daskalaki et al., 2004	Curriculum-based University Course Timetabling	Integer Programming	Optimization problem using 0-1 variables	University of Patras, Greece
Qualizza & Serafini, 2004		Integer Programming	Column Generation	University of Udine, Italy
Avella & Vasil'Ev, 2005		Integer Programming	Branch and Bound Algorithm	University of Sannio, Italy
Schimmelpfeng & Helber, 2007	Curriculum-based University Course Timetabling	Integer Programming		Hannover University, Germany
Bakir & Aksop, 2008	Mix of the two approaches	Integer Programming	Optimization problem using 0-1 variables	Gazi University, Turkey
Burke et al., 2012	Curriculum-based University Course Timetabling	Integer Programming	Branch and Bound Algorithm	University of Udine, Italy

Phillips et al, 2015	Curriculum-based Timetabling	University	Course	Integer Programming		University of Auckland, New Zealand
Colorni et al., 1992	Curriculum-based Timetabling	University	Course	Genetic Algorithm		Real world data, Italy
Khonggamnerd & Innet, 2009	Curriculum-based Timetabling	University	Course	Genetic Algorithm	Crossover, mutation and selection operators	University of the Thai Chamber of Commerce, Thailand
Alsmadi et al., 2011	Enrollment-based Timetabling	University	Course	Genetic Algorithm	Machine Learning	University of Jordan, Jordan
Rossi & Paechter, 2004				Memetic Algorithm	Local Search applied	ITC instances
Jat & Yang, 2008	Enrollment-based Timetabling	University	Course	Memetic Algorithm	Two Local Search applied	
Socha et al, 2003				Ant Colony	Ant Colony System and MAX-MIN Ant System	ITC instances

Nothegger et al., 2012	Enrollment-based Timetabling	University	Course	Ant Colony		ITC instances
Alvarez et al., 2002	Enrollment-based Timetabling	University	Course	Tabu Search	Local strategies to improve the timetable	Real world data, Spain
Cordeau et al., 2003				Tabu Search	Exchange moves applied	ITC instances
Aladag et al., 2009				Tabu Search	Four neighborhood structures	Hacettepe University, Turkey
Chen et al., 2020	Enrollment-based Timetabling	University	Course	Tabu Search	Move acceptance strategy	
Abdullah et al., 2005				Variable Neighborhood Search	Exponential Monte Carlo acceptance criteria	
Nguyen et al., 2011	Curriculum-based Timetabling	University	Course	Variable Neighborhood Search	Acceptance criteria applied	University of Science, Vietnam
Tuga et al., 2007	Enrollment-based Timetabling	University	Course	Simulated Annealing	Kemp Chain neighborhood used	Lewis instances

Aycan & Ayav, 2009		Simulated Annealing	Different types of neighborhood searching	Izmir Institute of Technology, Turkey
Ceschia et al., 2012	Enrollment-based University Course Timetabling	Simulated Annealing	Local search with six different stages	ITC and Lewis instances
Abdullah et al., 2007		Randomized Iterative Improvement with Composite Neighboring algorithm		ITC
Ross & Marín-Blazquez, 2005		Hyper-heuristic	Selection constructive	ITC instances
Qu & Burke, 2009	Enrollment-based University Course Timetabling	Hyper-heuristic	Selection constructive	Soncha instances
Rossi-Doria & Paechter, 2012	Enrollment-based University Course Timetabling	Hyper-heuristic	Selection constructive	Random instances
Kalender et al., 2012		Hyper-heuristic	Selection Perturbative	Yeditepe University, Turkey

Soria-Alcaraz et al., 2014	Enrollment-based University Course Timetabling	Hyper-heuristic	Selection Perturbative	ITC instances
Soria-Alcaraz et al., 2016	Mix of the two approaches	Hyper-heuristic	Selection Perturbative	ITC instances
Rattadilok, 2010		Hyper-heuristic	Generation perturbative	ITC instances
Lindah et al., 2018	Curriculum-based University Course Timetabling	Matheuristic	Mixed Integer Programming Model and Heuristics	ITC instances
Mikkelsen & Holm, 2022	Mix of the two approaches	Matheuristic	Mixed Integer Programming and parallelized heuristics	ITC instances
Jat & Yang, 2011	Enrollment-based University Course Timetabling	Hybrid Approach	Genetic Algorithm and Tabu Search	ITC instances
Kkohshori & Abadeh, 2012	Curriculum-based University Course Timetabling	Hybrid Approach	Randomized Iterative Local Search, Simulated Annealing and Tabu Search	

Goh et al., 2020	Enrollment-based Timetabling	University	Course	Hybrid Approach	Tabu Search with Sampling and Perturbation with Iterated Local Search	ITC instances
Rezaeipanah et al., 2021	Enrollment-based Timetabling	University	Course	Hybrid Approach	Parallel Genetic Algorithm with Local Search	ITC instances

Table A.3: Summary of UETP

Paper	Type of Problem	Solution Approach	Details	Application
Mirhassani, 2006	Uncapacitated UETP	Integer Programming	Two-Stage modeling approach	Shahrood University of Technology, Iran
Mccollum et al., 2012		Integer Programming		International Timetabling Competition, 2007
Cataldo et al., 2017	Curriculum-based approach	Three mathematical Programming Models		University of Diego Portales, Santiago of Chile
Al-Yakoob et al., 2010	Capacitated UETP	Mixed Integer Linear Programming		Kuwait University (KU), Kuwait
Reis & Oliveira, 1999	Curriculum-based approach	Constraint Programming	Constraint Satisfaction Techniques	University of Fernando Pessoa, Portugal

Merlot et al., 2002	Capacitated and Uncapacitated UETP	Constraint Programming	Simulated Annealing applied in a second phase	University of Melbourne, Australia
Huede et al., 2006,	Capacitated UETP	Constraint Programming	Branch and Bound	
Wong et al., 2002	Uncapacitated UETP	Genetic Algorithm	Fitness, selection, crossover, mutation and insertion operator	University of Québec, Canada
Pillay & Banzhaf, 2010	Uncapacitated UETP	Genetic Algorithm	Two phased approach	Carter et al. benchmarks
Innet, 2013	Uncapacitated UETP	Genetic Algorithm	Crossover, mutation and selection operator	University of the Thai Chamber of Commerce, Thailand
Jha, 2014	Uncapacitated UETP	Genetic Algorithm	Crossover, mutation and selection	Ibra College of Technology, Oman
Azimi, 2004	Uncapacitated UETP	Ant Colony Algorithm		

Eley, 2006	Uncapacitated UETP	Ant Colony Algorithm	MAX-MIN and ANTCOL approach	Toronto benchmark
Thepphakorn et al., 2014	Uncapacitated UETP	Ant Colony Algorithm	Best-Worst Ant System and the Best-Worst Ant Colony System	
Di Gaspero & Schaerf, 2000	Capacitated and Uncapacitated UETP	Tabu Search	Adaptive Tabu List	University of Nottingham, United Kingdom
White & Xie, 2000		Tabu Search	Four phase approach	University of Ottawa, Canada
DiGaspero, 2002	Capacitated and Uncapacitated UETP	Tabu Search	Multi-neighborhood local search	Carter et al. benchmarks
Bajeh & Abolarinwa, 2011		Tabu Search		
Burke et al., 2010	Capacitated and Uncapacitated UETP	Variable Neighborhood Search	Genetic Algorithm used	Carter et al. benchmarks

Alefragis et al., 2021		Variable Neighborhood Search	Simple and complex moves	Toronto benchmarks
Thompson & Dowsland, 1996	Uncapacitated UETP	Simulated Annealing	Kempe chain neighborhood	University of Wales,, United Kingdom
Thompson & Dowsland, 1998	Uncapacitated UETP	Simulated Annealing	Two-phase approach	University of Wales, United Kingdom
Leite et al., 2019	Capacitated University UETP	Simulated Annealing	Fast Simulated Annealing	ITC instances
Bellio et al., 2021	Uncapacitated UETP	Simulated Annealing	Combination of multiple neighborhoods	Carter et al. instances
Sabar et al., 2012	Uncapacitated UETP	Hyper-heuristic	Selection Constructive hyper-heuristic	Carter et al. instances
Soghier & Qu, 2013	Capacitated University UETP	Hyper-heuristic	Selection Constructive hyper-heuristic	ITC instances

Sin & Kham, 2012	Uncapacitated UETP	Hyper-heuristic	Selection perturbative hyper-heuristic	Carter et al. instances
Anwar et al., 2013	Capacitated University UETP	Hyper-heuristic	Selection perturbative hyper-heuristic	ITC instances
Pillay & Banzhaf, 2009	Uncapacitated UETP	Hyper-heuristic	Generation constructive hyper-heuristic	Carter et al. benchmarks
Pais & Burke, 2010	Uncapacitated UETP	Hyper-heuristic	Generation constructive hyper-heuristic	Carter et al. benchmarks
Rahman et al., 2014		Hyper-heuristic	Generation constructive hyper-heuristic	
Qu et al., 2009		Hybrid Approach	Largest Weighted Degree with Saturation Degree	
Alzaqebah & Abdullah, 2015		Hybrid Approach	Hybrid Bee Colony Optimization	Uncapacitated University UETP

Table A.4: Summary of TDTP

Paper	Solution Approach	Details	Application
Huynh et al., 2012	Genetic Algorithm	Crossover and mutation operators	Hanoi University of Science and Technology, Vietnam
Kochanikova & Rudova, 2013	Local Search		Masaryk University, Czech Republic
Dung et al., 2015	Constraint-based Local Search	Java library used	Hanoi University of Science and Technology, Vietnam
Fastre et al., 2017	Mixed Integer Programming		Louvain School of Engineering, Belgium
Battistutta et al., 2019	Integer Programming, Constraint Programming and Simulated Annealing		University of Bicocca, University of Milan, and University of Modena and Reggio Emilia, Italy
Christopher & Wicaksana, 2021	Particle Swarm Optimization		University Multimedia Nusantara, Indonesia