An Integer Programming Model for Scheduling Master's Thesis Defences

João António Carvalho Almeida

Dissertation to obtain the Master of Science Degree in

Industrial Engineering and Management

Supervisor(s): Prof. Inês Marques
Dr. Daniel Rebelo dos Santos

Examination Committee

Chairperson: Prof. Tânia Pinto Varela
Supervisor: Dr. Daniel Rebelo dos Santos
Member of the Committee: Prof. Ana Sara Costa

September 2021

Abstract

In the Department of Engineering and Management at Instituto Superior Técnico, the master's thesis defence scheduling is the responsibility of the department's secretary. The aim of this work is to automate that process. The problem is formulated as a mixed integer linear programming model, with four objectives. The first maximises the number of defences to be scheduled. The second prioritises the satisfaction of individual preferences, the third minimises the number of times committee members must travel to the Taguspark *Campus* and, lastly, the fourth promotes the compactness of the schedules. Two different approaches to solve the model are introduced, the Two Stage *a Priori* Approach and the Two Stage Augmented ϵ - Constraint Approach. Both include a first stage where the maximum number of thesis defences that can be scheduled is found and then set as a hard constraint for the second stage. The second stage is where the approaches differ, with the *a Priori* Approach including the remaining three objectives in a single weighted objective function and presenting one optimal solution accordingly and the Augmented ϵ - Constraint Approach presenting several Pareto Optimal solutions. The usefulness of the first stage is proven in the computational experiments, as instances where not all thesis defences could be scheduled appeared. Furthermore, even in the largest tested instances, this stage never took more than one minute to solve. As for the second stages, the Augmented ϵ - Constraint Approach takes considerably longer times to solve than its counterpart, which is the trade-off for being able to know more possible solutions of the Pareto set.

Keywords— Thesis Defence Scheduling , Academic Scheduling, Preference Modelling, Fairness, Integer Programming, Multi-objective, ϵ - constraint method

Resumo

No Departamento de Engenharia e Gestão do Instituto Superior Técnico a calendarização das defesas de tese é responsabilidade da secretária do departamento. O objectivo desta dissertação é automatizar esse processo. O problema é modelado como um modelo de programação linear inteira mista, com quatro objectivos. O primeiro maximiza o número de defesas calendarizadas, o segundo a satisfação de preferências individuais, o terceiro minimiza o número de dias em que um membro do júri está escalonado para uma defesa e o quarto promove a ocorrência de horários compactos. Dois métodos de resolução do modelo são propostos. O primeiro nomeado Two Stage *a Priori* Approach e o segundo Two Stage Augmented ϵ - Constraint Approach. Ambos os métodos começam com uma primeira fase igual, em que o primeiro objectivo é maximizado e o seu valor passará como uma restrição para a segunda fase. Após este ponto, no primeiro método, os restantes três objectivos são aglomerados numa função-objectivo ponderada e um único resultado é apresentado. No segundo método, vários resultados da fronteira de Pareto são apresentados. A utilidade da primeira fase é provada à medida que aparecem instâncias em que parte das defesas não pode ser calendarizada. Em relação às segundas fases, o primeiro método foi muito mais rápido que o segundo, com a contrapartida de o segundo proporcionar um melhor conhecimento da fronteira óptima.

Palavras-Chave— Calendarização de Defesa de Teses de Mestrado, Modelação de Preferências, Equidade, Programação Linear, Múltiplos Objectivos, Método ϵ - Constraint

Contents

List of Figures

List of Tables

Acronyms

DEG Department of Engineering and Management.

IST Instituto Superior Técnico.

MEGI Master's Degree in Industrial Engineering and Management.

MILP Mixed Integer Linear Programming.

OECD Organisation for Economic Co-operation and Development.

Chapter 1

Introduction

In this chapter, the aim and motivation of this dissertation is presented in Section 1.1. Secondly, a set of objectives that will ultimately lead to the resolution of the problem at hand is proposed in Section 1.2. Finally, an outline of the dissertation is given in Section 1.3.

1.1 Motivation

Thesis defence scheduling is a recent branch within the academic scheduling field which has recently been receiving increased attention. After being first introduced by Huynh et al. (2012), several other works have contributed to the literature on this problem. Nonetheless, it is still evident that the research on this topic is not as extensive as it is for other academic scheduling branches, namely, exam scheduling and course timetabling. Thus, there is space to study and add to the literature on this field by comparing it to the developments that have been done in its academic scheduling counterparts and applying it to different instances.

The motivation for previous research in this area has been the inefficient and time-consuming methods that are usually employed to schedule the thesis defences at universities. This process is usually done by one person who has to aggregate the availability and requests of all the examination committee members and produce a fitting schedule by hand. Depending on the number of defences to be held, this results in a tedious process that does not always produce satisfactory results to all the affected. Thus, by implementing optimization methods to this problem, it has been possible not only to lighten the workload for the people responsible for scheduling these events but also to produce better quality schedules for the committee members.

The same reasons apply to the case at hand, where the scheduling process for the thesis defences in the Department of Engineering and Management (DEG) at Instituto Superior Técnico (IST) is the responsibility of the department secretary, who has to mediate the scheduling of each thesis defence between the members of the examination committee, usually a chairperson, a supervisor and an additional member. Furthermore, a portion of the committee members are not professors at the *campus* where these defences are held, meaning that they must move from their usual working places to be present, thus, respecting their requests is paramount, and not always possible to do.

Accordingly, the aim of this thesis is to automate this process at the DEG at IST, which schedules the defences for the Master's Degree in Industrial Engineering and Management (MEGI), by proposing an optimization method which is fair for all the examination committee members, satisfies their individual requests and reduces the number of times they need to move themselves to the Taguspark *Campus*, while promoting compact schedules for committee members, allowing for better quality schedules than the current process and reducing the workload of the department secretary. If the resulting method is deemed viable and produces satisfactory results for the scope of the problem, it should be possible to adapt the model to other departments of the university.

1.2 Objectives

After contextualizing and presenting the motivation for this dissertation, it is fundamental to define the objectives that will ultimately lead to the solution of the problem at hand.

As it was previously stated, the general goal of this work is to automate the process of scheduling master thesis defences for the MEGI and lay the groundwork for a possible university-wide solution for this problem.Thus, several intermediate objectives will have to be set in order to successfully achieve the proposed target:

• Clarify the regulations regarding master degrees and master thesis defences in Portuguese universities;

• Introduce the university (IST) and clarify its rules regarding master's thesis defences, which regulate aspects such as deadlines and examination committee compositions;

• Introduce the university department whose scheduling process will be the focus of this dissertation (DEG) and describe in further detail its current thesis defence scheduling process. Concretely, by mentioning aspects such as examination committee composition, number of thesis defended each semester and steps taken to reach the final schedules;

• Understand the progress that has been done in the literature regarding thesis defence problems and comparing it to other, more common, academic scheduling problems, namely exam scheduling and course timetabling;

• Identify important aspects for the case at hand that may not yet be fully developed in the thesis defence literature and explore further literature on academic scheduling and other fields that address these issues;

• Identify the necessities and priorities of the decision-maker and understand which objectives should be focused on;

• Propose a Mixed Integer Linear Programming Model that accounts for all the characteristics of the problem at hand and empowers the decision-maker to find solutions according to their preferences;

• Formulate algorithms that allow the model to be solved efficiently;

• Understand the computational capacity the algorithms and model need;

• Explore the effect that varying select characteristics of generated instances has on the efficiency of the model;

• Provide future users of the tool with a comprehensive user guide that allows them to schedule thesis defences and easily import all the necessary data.

1.3 Thesis Outline

This dissertation is organized in the following seven chapters:

1) Introduction

The motivation for this work is presented and a set of objectives are proposed in order to solve the previously introduced problem.

2) Problem Contextualization

The regulations that oversee the thesis defences in Portugal are addressed. Furthermore, the university in question, IST, is presented, with a special focus on its master thesis defence guidelines as well as the current scheduling method used at the DEG.

3) Literature Review

A bibliographic review is made on articles regarding both thesis defence scheduling and other branches of the academic scheduling field, namely exam scheduling and course timetabling, as well as on other concepts deemed important for this work, specifically preference modelling and fairness in scheduling.

4) Mathematical Model

This chapter proposes a Mixed Integer Linear Programming Model formulation for the problem at hand, including sets, parameters, variables, constraints and objectives.

5) Problem Solution

Two different algorithms to solve the aforementioned model are proposed. The first one is based on a weighted single objective formulation where only one solution is provided whereas the second considers a multi-objective formulation that proposes several Pareto optimal solution to each instance.

6) Computational Experiments

A instance generator that outputs realistic instances is proposed, and several computational experiments are made on those instances so that the efficiency of the algorithms may be assessed.

7) Conclusions and Future Work

The overall conclusions of the dissertation are summarised and a direction for future research is proposed.

Chapter 2

Problem Contextualization

To obtain a master's degree, besides from having obtained a set number of ECTS credits, students at IST are required to produce a thesis which must also be defended before an examination committee. As part of the University of Lisbon, IST must comply with the rules set forth by the government for awarding degrees to its students, which, in this case, are currently regulated by the *Decreto-Lei n.º 65/2018 de 16 de Agosto do Ministério da Ciência, Tecnologia e Ensino Superior* (2018).

The present chapter will present a contextualization of the problem at hand. Firstly, an overview of the implications the aforementioned decree-law has on the thesis defences at IST is presented in Section 2.1 as well as an introduction of the institution with a focus on the present situation regarding thesis defences in Section 2.2. Secondly, the specific case of the DEG is addressed, with a presentation of the department and of its thesis scheduling process being made in Section 2.3. Finally, in Section 2.4, a summary of the rest of the chapter is presented.

2.1 Master's Thesis Defences in Portugal

Between 2016 and 2017, by request of the Portuguese government, the Organisation for Economic Co-operation and Development (OECD) proceeded to the evaluation of the country's higher education and scientific innovation systems, and, in February 2018, the organization presented a set of recommendations to improve them (*Decreto-Lei n.º 65/2018 de 16 de Agosto do Ministério da Ciência, Tecnologia e Ensino Superior* 2018, OECD 2018). Then, based on those, a decree-law was emitted. This decree-law introduced several changes to the rules for obtaining academic degrees and higher education diplomas, nonetheless, the guidelines for master thesis defences were not the main focus of the OECD evaluation, thus, they remained relatively unchanged with the emission of the aforementioned document.

The study cycle for a master's student in Portuguese universities can go from 90 to 120 ECTS credits, with a normal duration of three to four curricular semesters. However, in some cases, where the curriculum is more focused on a professional orientation, students can obtain their master's degree with only 60 ECTS credits. Furthermore, the study cycle must include a specialization course, composed by several course units, to which at least half of the ECTS credits are assigned and a scientific thesis or professional internship, with an assignment of at least 30 ECTS credits and that must be supervised by a doctor or a recognized specialist in the field.

The thesis or the final internship report must then be defended before an examination committee, which must include between three and five members, assigned by the university and which may include the student's supervisor. In cases where the study cycle is divided between a Portuguese and foreign institution, and the student has two supervisors, the committee can include both of them, being constituted by five to seven members instead. Furthermore, the institutions can, as long as they comply with the guidelines ascribed by the decree-law, define their own sets of rules which can vary between them.

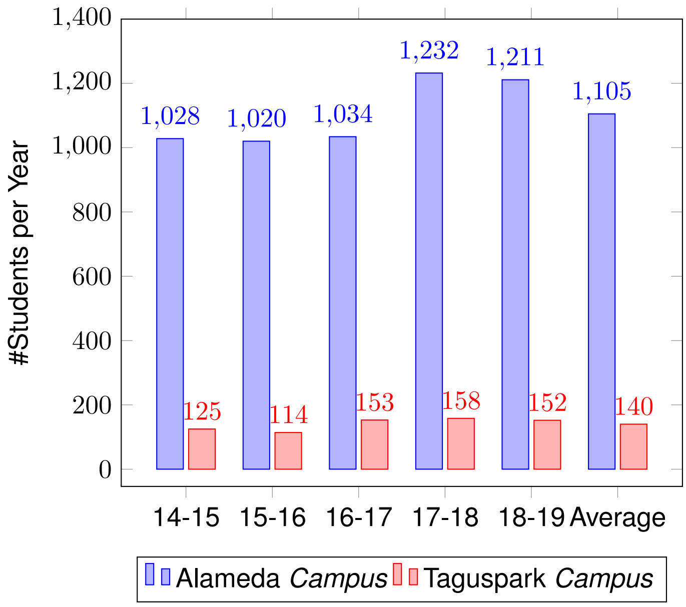
2.2 Master's Thesis Defences at Instituto Superior Técnico

IST, incorporated in the University of Lisbon, is the largest school of engineering and technology in Portugal, with approximately 11000 enrolled students in either bachelor's or first study cycle integrated master's (48 % of total students), master's or second study cycle integrated master's (42 % of total students) or doctorate degree courses (10 % of total students) (*Balanço RAIDES 2019-20* 2019).

Amongst all the study cycles, in the 2019-2020 school year there were 84 different degree courses in IST (*Balanço RAIDES 2019-20* 2019). Each individual degree is assigned to at least one of its ten departments. Furthermore, there are two different *campi* where each degree's course units can be allocated and a third one solely for research purposes. The Alameda *Campus* is the oldest IST *campus*, and it is the one where the majority of the students and teachers are allocated, with an average of approximately 9500 enrolled students over the last five years (*Balanço RAIDES 2019-20* 2019). The Taguspark *Campus*, is the more recent one, with its construction having finished in 2009, and it is located roughly 20 kms away from the Alameda *Campus*. The transportation between the two *campi* is assured by IST via an hourly shuttle between them, which also passes through a number of alternative stops. In the last five years there has been an average of approximately 1500 enrolled students in this *campus* (*Balanço RAIDES 2019-20* 2019), nonetheless it is expected that it may house a higher number in the future.

For the last five years, as we can see in Figure 2.1, on average, 1245 students finished their second study cycle degrees at IST, which includes both master's degrees and the second study cycle of integrated master's degrees. Out of those, approximately 89 % were enrolled on the Alameda *Campus* whereas the remaining ones are from the Taguspark *Campus*. To be awarded a master's degree in IST, in accordance with the Portuguese Law, students must conclude and defend, before an examination committee, a scientific thesis regarding a theme related to their degree. IST has a set of rules in place that affect these research works, namely by setting the timeline for the several steps that lead to the elaboration of the thesis as well as the composition of the examination committees, which must be in concordance with the Portuguese guidelines. Furthermore, each thesis defence must be scheduled, a process which IST leaves for each department to handle as they see fit.

According to the Portuguese (*Decreto-Lei n.º 65/2018 de 16 de Agosto do Ministério da Ciência, Tecnologia e Ensino Superior* 2018) and IST regulations (*Regulamento de Unidades Curriculares de*



Source: *Balanço RAIDES 2019-20* (2019)

Figure 2.1: Number of Master's Students Finishing Their Degree at IST per Year

Dissertação (Mestrado) 2020), for a defence to be held, besides the defendant, at least three committee members must be present. Firstly, there is the chairperson, who can either be the degree coordinator or a member of the scientific committee that the former appoints. Furthermore, the supervisor of a thesis cannot be appointed as the chairperson for that same thesis. Secondly, there is the supervisor, who has guided the defendant through the execution of the thesis and, if there are two or more supervisors, only one of them can be part of the committee and, lastly, there is an additional member, who is appointed by the advisor to argue with the defendant and that can be either national or foreign.

However, before students can start working and eventually conclude their thesis, they must be assigned a theme. The dates and deadlines for the steps that lead to the theme assignment are set by IST (Table 2.1) and usually remain relatively unchanged from year to year. Every year, there are two phases where students can be assigned to a theme, one in the first semester and one in the second. When students are assigned a theme in one semester, they usually start working on it in the following one. Nonetheless, due to the 2020 global pandemic, some of these dates have been updated, even so, for the purpose of this work, the dates published prior to that situation will be considered.

Firstly, a set of professors is invited by the scientific committee of each course to submit proposals for thesis themes, a step which they will have at least one month to do. Afterwards, the list of possible supervisors and themes is published via the IST portal (FÉNIX), and then, for about a month, students can apply to a set number of themes. Lastly, after the application deadline, the scientific committee and the possible supervisors, will have roughly five weeks in the first phase and two in the second to assign the themes to students. It is also a possibility that students propose themselves a theme to a professor, who can then accept or not the student proposal.

Just like it was for theme assignment, IST has two sets of deadlines regulating the actions that will

Table 2.1: Dates for Submission, Application and Assignment of Dissertation Themes

1st Phase Dates	Steps	2nd Phase Dates
16 September	Beginning of the semester	17 February
30 September	Invitation to submit thesis proposals	31 March
30 October	Dissertation proposals submission deadline	30 April
15 November	Themes and supervisors publication	15 May
15 December	Application for thesis themes deadline	15 June
23 January	Themes assignment deadline	30 June

Source: *Tramitação da Dissertação/Projeto (Mestrado)* (2020), *Calendário Escolar 2019/2020* (2020)

lead to the conclusion of the evaluation process for its master students, one for those who concluded their work in the first semester and another for the ones who did it in the second (Table 2.2). These have remained relatively unchanged over the last few years, except for a few updates for the same reasons discussed previously.

After the conclusion of their work, students have a deadline until which they must submit their thesis to their course coordinator or the scientific committee in cases where the former is the supervisor. Afterwards, the course coordinator must approve the examination committee for each discussion within a period of two weeks after the previous deadline. The defence of each thesis must then be scheduled and realized within a one-month window. Ensuing this discussion, students will have one week to submit all the necessary documentation via FÉNIX. Two weeks after the deadline for all defences, the course coordinator or the scientific committee in cases where the former is the supervisor must verify all documents and, two weeks later, comes the deadline for the issuing of all the grades.

Table 2.2: Deadlines for Dissertation Submission, Defence, Committee Approval and Evaluation

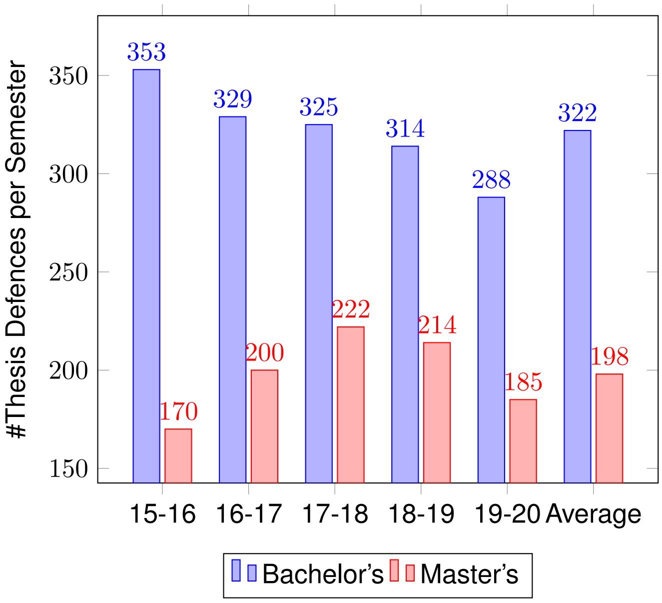
1st Semester Dates	Steps	2nd Semester Dates
16 September	Beginning of the semester	17 February
15 May	Dissertation submission	15 October
31 May	Examination committee approval	31October
30 June	Dissertation defence	30 November
7 July	Final version thesis submission	7 December
15 July	Verification of the submitted documents	15 December
31 July	Grades Issuing	31 December

Source: *Tramitação da Dissertação/Projeto (Mestrado)* (2020), *Calendário Escolar 2019/2020* (2020)

2.3 Master's Thesis Defences at the Department of Engineering and Management

The Department of Engineering and Management (DEG) is the most recent of the ten departments existent in IST and, as it was previously stated, its master thesis defence scheduling processes are the main focus of this thesis.

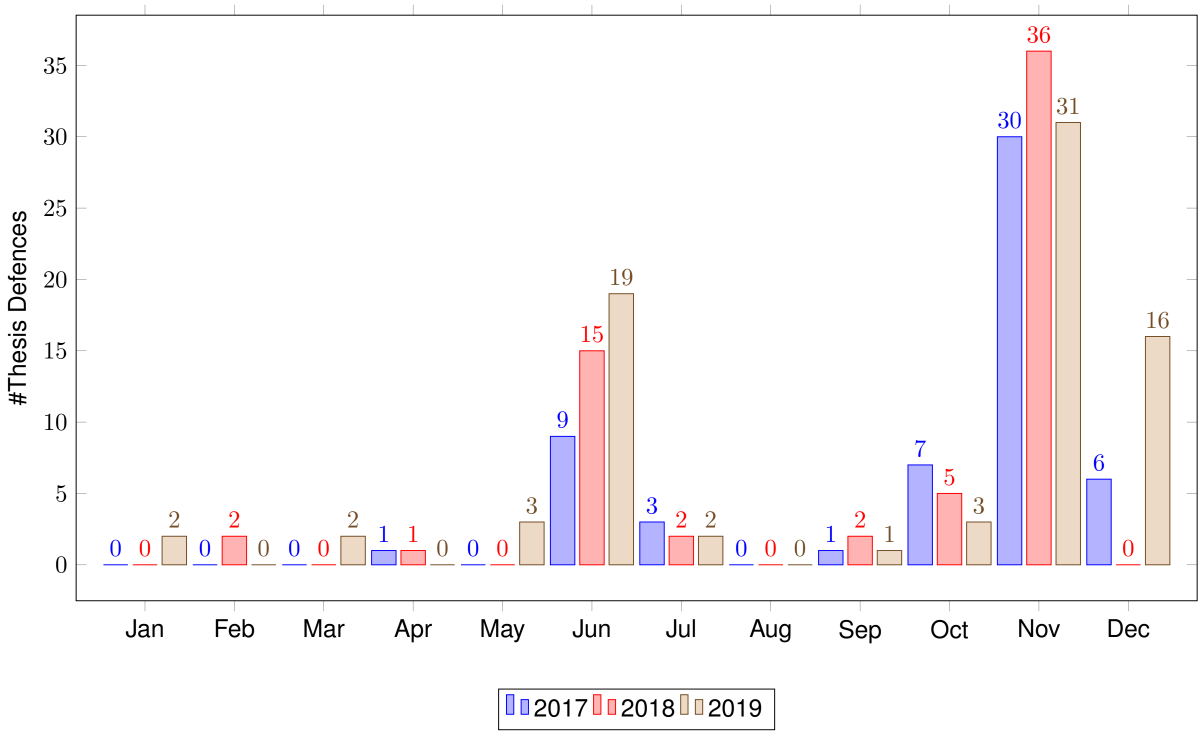
The DEG has offices in both the Alameda and Taguspark *campi* and oversees not only research work but a number of course units as well. The department is responsible for both the Bachelor and Master Degree in Industrial Engineering and Management at IST which have had an average of approximately 322 and 198 students in the last five years respectively (Figure 2.2) as well as a set of doctorate degree courses within the scope of the department. While the research work and the third study cycle degree courses are often based in the Alameda *Campus*, the majority of the activities related to the master and bachelor degrees, that is, classes, presentations, exams, thesis discussions, amongst others, are undertaken in the Taguspark *Campus*.



Source:*Balanço RAIDES 2019-20* (2019)

Figure 2.2: Number of Industrial Engineering and Management Students per Study Cycle

Between the 2016-17 and the 2018-19 school years on average 20 and at most 28 MEGI students have had their thesis discussions in between January and July, and on average 46 and at most 51 have done so between August and December (Figure 2.3). Furthermore, in both semesters, we can see a tendency for most of the thesis to be submitted close to the deadline (Table 2.2), meaning that between the submission and the deadline for the scheduling of the defences there are roughly five or six weeks when they can be scheduled, leading to a concentration of most discussions in the month of June, with an average of 20% of the defences per year, and November where 50% of the defences are scheduled (Figure 2.3).



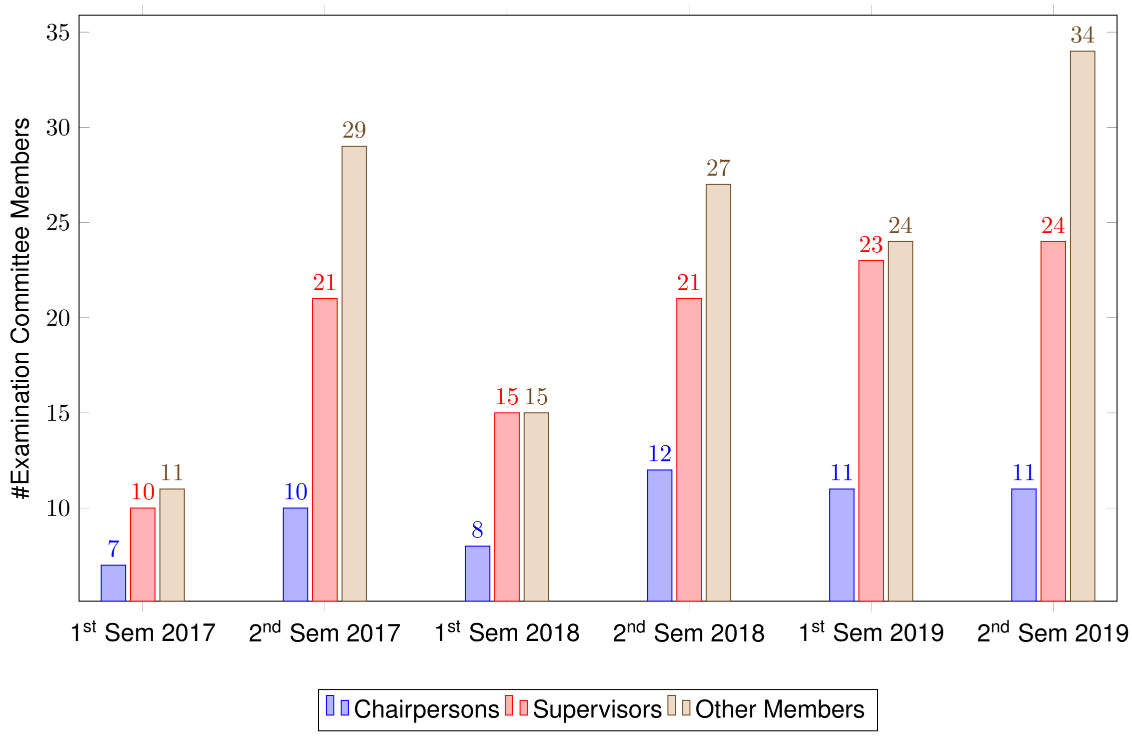
Source: *Master's in Industrial Egineering and Management Dissertations at Instituto Superior Técnico* (2020)

Figure 2.3: Number of MEGI Thesis Defences at IST per Month and Year

In the DEG, the scheduling of all the discussions is delegated to the department's secretary, who, as she receives the indication that a thesis has been submitted and must be defended, has to collect the availability of the examination committee members and moderate the scheduling process between them as they converge to a date that suits their needs. However, since this effort is not coordinated between all the committees, this ends up being difficult to achieve and some professors end up being favoured, while others have to accept less desirable dates for their panels.

Currently, the process for scheduling the defences consists of three steps. Firstly, the chairpersons, who must be constituents of the DEG scientific committee, are asked their availability. There has been an average of 10 per semester over the last years (Figure 2.4). Afterwards, a doodle is sent to the chairperson and the remaining examination committee members, including the supervisors, which have been on average 19 per semester over the last three school years (Figure 2.4), and other members, who have been an average of 23 per semester (Figure 2.4). In total, in the 2016-17, 2017-18 and 2018-19 school years, 55, 51 and 71 different committee members were involved in thesis discussions, respectively (*Master's in Industrial Egineering and Management Dissertations at Instituto Superior Técnico* 2020).

After having the dates for the discussions, the DEG secretary has to send them to the Taguspark Facility Management Unit, which is a service for the Taguspark *Campus* responsible for managing a set of activities related to the building. The service must then assign one of five possible rooms to each defence, while trying, as long as possible, to ensure that when chairpersons have several discussions in a row, that they can remain in the same room.



Source: *Master's in Industrial Egineering and Management Dissertations at Instituto Superior Técnico* (2020)

Note: In the 2016-17, 2017-18 and 2018-19 school years 1, 1 and 18 different thesis, respectively, did not disclose the identity of the chairperson and other member

Figure 2.4: Number of Examination Committee Members per Position and Semester in MEGI Thesis Defences

Due to the fact that the regular offices for a considerable number of examination committee members are in the Alameda *Campus* and that the two main paths between the two *campi*, through the highways A5 and A37 are amongst the four worse regarding traffic in Lisbon (Schrank et al. 2015), the professors generally prefer to move themselves the least possible to the Taguspark *Campus*. The chairpersons are the ones who generally have to be present for the most discussions, with an average of 5.70 discussions per year per chairperson, in contrast to the 2.26 per supervisor and 1.74 per additional member, disregarding their presence in other roles within the examination committee (Figure 2.3 and Figure 2.4). Furthermore, if the presence in different positions is taken into account, it is not uncommon that some chairpersons have to be present in more than 10 different discussions every year (*Master's in Industrial Egineering and Management Dissertations at Instituto Superior Técnico* 2020). Thus, there is usually an attempt to guarantee that their discussions are spread over a period of only one or at most two days within each semester. Then, whenever possible, there is an attempt to minimise the movements between *campi* that other members of the committee who must attend multiple discussions.

2.4 Chapter Considerations

IST, as part of a Portuguese university, must adhere to the country's regulations regarding second study cycle degree courses and thesis discussions. These are all currently described in the *Decreto-Lei n.º 65/2018 de 16 de Agosto do Ministério da Ciência, Tecnologia e Ensino Superior* (2018), which has been emitted after an assessment of the Portuguese higher education system, requested by the government to the OECD.

Thus, IST has a set of rules, in compliance with the Portuguese Law, that oversees the whole process for obtaining a master's degree and conducting a thesis in the university. Each year, a timeline for all the necessary steps to undertake is published, going from the theme assignment to the final discussion and grade issuing. Furthermore, the general rule at IST is that the examination committees are composed by three members: a chairperson, who must be part of the department's scientific committee; the thesis supervisor, who must be a professor or researcher at the university or a specialist recognized by the scientific committee and who guides the students through the execution of their thesis and, finally, one or more additional members who, just like the supervisor, must be professors or researchers at the university or specialists recognized by the scientific committee.

This work is focused on the DEG processes for scheduling master thesis defences, with the expectation that it might be a gateway for a possible university-wide solution. This is the most recent department at IST as of the writing of this thesis, and it oversees both research work and the bachelor's and master's Degree in Industrial Engineering and Management at the institution, as well as a few other doctorate degree courses.

In the department in question, the secretary oversees the scheduling of all the thesis discussions of the MEGI. This is a time-consuming task where it is not possible to always comply with the preferences of all the affected, specially considering that all the defences take place at the Taguspark *Campus*, whereas a good portion of the professors has their regular offices in the other *campus*. Thus, they usually schedule their examination committees while trying to reduce their movements, by minimizing the number of days when they must be present for a discussion.

To get a better grasp on the problem at hand, a detailed literature review on other works related to thesis defence scheduling will be made as well as other academic scheduling problems, as this will lead to a better understanding of how aspects of the problem such as the objectives or hard constraints have been dealt with before. Furthermore, since the problem at hand is so closely related to the individual preferences of the professors, which up to this point have been assured as much as possible by them while scheduling the thesis discussions, it is also important to understand how other works inside and outside the academic scheduling field have dealt with this issue. Lastly, since there is a concern regarding the fairness of the different schedules assigned to each professor, the following review should also give an insight on how this aspect can be guaranteed.

Chapter 3

Literature Review

The Thesis Defence Scheduling problem addressed in this work aims to automate the process of scheduling thesis defences, reducing the workload of the staff previously responsible for this task, while producing schedules that better suit the availability of the examination committee members. This problem has not been extensively explored in the literature, thus, other problems within the academic scheduling field are analysed within this review, namely exam scheduling and course timetabling. In the aforementioned field, one of the main concerns is to create timetables/schedules that better suit the preferences of the involved. Thus, further review on the topic of preference modelling is also made, as well as an analysis on scheduling articles that have a focus on fairness, to understand how this issue is usually dealt with. Sections 3.1, 3.2 and 3.3 regard the literature on thesis defence scheduling, exams scheduling and course timetabling, respectively. An overview of the methods used to solve the problems is presented, as well as the most common hard constraints and objectives. Finally, these are compared with the case at hand, the thesis defence problem at IST. Section 3.4 addresses preference modelling, and Section 3.5 analyses fairness in scheduling problems. Finally, in Section 3.6 a summary of the full chapter is presented.

3.1 Thesis Defence Scheduling

The Thesis Defence scheduling is a recent branch within the academic scheduling field which also includes timetabling and exams scheduling. Since not many works on the subject exist, a more systematic approach to find related literature was not possible. Instead some of the papers analyzed in this section were found by searching "thesis defence scheduling" on google scholar and the others by searching for papers that would cite the former.

A few different approaches have been taken to tackle this problem, namely genetic algorithm (Huynh et al. 2012), local search (Kochaniková & Rudová 2013, Dung et al. 2015), optimization models (Fastré 2017, Battistutta et al. 2019) and simulated annealing (Battistutta et al. 2019).

There are some similarities between the scheduling process for all universities, leading to universal constraints present in all the literature so far, specifically:

• Every master thesis defence must be scheduled;
• The examination committee members cannot be present in two defences at the same time;
• Rooms cannot hold more than one defence at the same time;
• The assigned professors must be available at the time of the defence.

In addition to the aforementioned constraints all universities have different examination committee compositions that the models must take into consideration. In most cases, the examination committee of each thesis is composed by a set number of members, usually three (Kochaniková & Rudová 2013, Fastré 2017) or five (Huynh et al. 2012, Dung et al. 2015), with their respective functions. Nonetheless, in the case of the University of Udine (Battistutta et al. 2019), this number may vary from seven up to ten according to some rules. For all the stated cases, there are some members of the examination committee that must attend some defences, and the rest is chosen by the proposed models. Typically, some examination committee members are assigned to individual time slots for the defences that they must attend (Huynh et al. 2012, Kochaniková & Rudová 2013, Dung et al. 2015, Fastré 2017), however, there are also cases where some members of the examination committee are assigned to complete sessions of several defences instead (Kochaniková & Rudová 2013, Fastré 2017) and one case where all the examination committee members are assigned through this method (Battistutta et al. 2019).

The most common objectives are reducing or evening out the total workload of the professors Kochaniková & Rudová (2013), Dung et al. (2015), Fastré (2017), Battistutta et al. (2019) and creating more compact schedules for the examination committee members (Huynh et al. 2012, Kochaniková & Rudová 2013, Fastré 2017). Furthermore, the suitability between the theme of a thesis and the examination committee assigned to it is considered in some of them (Huynh et al. 2012, Dung et al. 2015, Battistutta et al. 2019), as well as reducing the changes in the classrooms(Huynh et al. 2012) and adding a reserve day when defences should not be scheduled (Fastré 2017).

The number of students varies between the cases, ranging from a minimum of 9 (Huynh et al. 2012, Dung et al. 2015) to a maximum of 551 (Battistutta et al. 2019). Nonetheless, the ratios between the number of professors and defences to be scheduled, show little variance, normally ranging between 1 and 2 with some exceptions. The timespan for the scheduling of all defences is usually less than one week, except for the University of Udine (Battistutta et al. 2019) case where it can go up to 33 days.

Due to size disparity as well as different objectives and examination committee composition it is not possible to directly compare most cases with each other to understand which approaches are the most efficient at solving the problem, nonetheless in (Battistutta et al. 2019) three different models were proposed in order to get a better grasp of this issue. The authors concluded that constraint programming was not a good fit for their problem, and that integer programming would outperform other methods with real world samples.

To account for the possibility of the non-existence of a feasible solution two different models are proposed by Fastré (2017). The first one and the one that is preferably chosen states that all thesis must be scheduled, the second one is only used if the first one fails to provide a feasible solution, and it includes the additional objective of minimising the number of defences that go unscheduled, while assigning a high enough penalty for every case where a thesis would go unscheduled that this only

Table 3.1: Summary of the Literature on Thesis Defence Scheduling Problems

	Solution Method				Examination Committee Characteristics													Professor to Student Ratio
						Assignment		Objectives							Schedule			
Paper	GA	LS	IP	SA	Size	Preset	Mixed	1)	2)	3)	4)	5)	6)	7)	Division	TimeSpan		
Huynh et al. (2012)	✓				5		✓		✓	✓		✓		✓	Time Slots			1.5-7
Kochaniková & Rudová (2013)		✓			3		✓	✓		✓					Time Slots	4 days		0.8-1.2
Dung et al. (2015)		✓			5		✓	✓	✓						Time Slots			1.4
Fastré (2017)			✓		3		✓			✓	✓			✓	Time Slots	3 days		1.5-1.9
Battistutta et al. (2019)			✓	✓	7-10		✓	✓	✓						Sessions	2-33days		0.5-2.4
Present Work			✓		3	✓				✓			✓	✓	Hours	1 month		0.8-0.96

Solution Method: GA) Genetic Algorithm; LS) Local Search; IP) Integer Programming; SA) Simulated Annealing

Objectives: 1) Minimise or even out workload for professors; 2) Maximise the suitability between the subject of the thesis and the assigned examination committee; 3) Compactness of the schedule; 4) Minimise the number of thesis scheduled in the reserve day; 5) Minimise the overlap between a session that a professor must attend with their busy time; 6) Personalised individual professor preferences; 7) Minimise the unscheduled thesis defences happens as a last resort. The work by Huynh et al. (2012) also uses a model similar to the second one.

There are a few similarities between the cases explored in the previous literature and the one at IST. As a thesis defence scheduling problem, the hard constraints should be similar to the ones already seen, while considering the different examination committee structure and time division. Furthermore, the committee composition is close to the ones in some examples in the literature (Kochaniková & Rudová 2013, Fastré 2017). Nonetheless, one of the differences is that in our case the committee is already fully predetermined and provided by the supervisor. Another difference between the case at IST and the cases in the literature is that there are no predefined sessions or time slots for the scheduling of defences, and each defence can be assigned to any hour. One area are where there is room for new objectives is the personal preferences of the professors, which up to this point are only represented as periods of unavailability or availability. One of the cases (Fastré 2017) adds another layer to this issue by including a reserve day when defences should not be scheduled, but this is not flexible enough for the case at IST.

As exam and timetable scheduling are problems that are both academic scheduling problems that have been extensively studied, in contrast to the thesis defence scheduling problem, an analysis of both of these fields might bring some insight into what kind of issues might have been overlooked by the literature up to this point.

A summary of the literature analysed in this section is provided in Table 3.1 including a comparison to this work.

3.2 Exams Scheduling

Exams scheduling is one of the original branches in academic scheduling, thus, it has been more explored than master thesis defence scheduling. Therefore, only a portion of the literature was analysed. All the articles were searched using Web of Science, under the search topics "Exam Scheduling" and "Exam Timetabling" and the 17 most cited papers from 2007 to the present day, which were not literature reviews or unrelated and were available through the IST VPN service, were selected.

Just like for thesis defence scheduling a variety of approaches have been applied to this problem, such as ant colony algorithms (Eley 2007), hyper-heuristics (Burke et al. 2007, Sabar, Ayob, Qu & Kendall 2012), graph colouring algorithms (Sabar et al. 2009), local search (Caramia et al. 2008), evolutionary algorithms (Cheong et al. 2009), hybrid heuristics (Qu et al. 2009), constructive heuristics (Kahar & Kendall 2010), pattern recognition search (Li et al. 2012), variable neighbourhood search (Elloumi et al. 2014), great deluge(Kahar & Kendall 2015), column generation algorithms (Woumans et al. 2016), hill climbing algorithms (Burke & Bykov 2017) binary programming (Wang et al. 2010, Cavdur & Kose 2016), tabu search based algorithms (Amaral & Pais 2016) or simulated annealing (Leite et al. 2019).

The characteristics of this problem lead to some similarities regarding hard constraints that are common to most papers, such as:

- No student can take two exams at the same time;

• The rooms that hold the exams must have enough seating capacity;

• All the exams are scheduled once;

• There must be enough available time for realization of the entire exam.

Furthermore, the elimination of days when students have two or more exams in the same day or consecutive days is another common aspect which is often considered. Due to shorter available time span, number of students or other reasons, it is not always possible to ensure that the schedules comply with this rule, thus in some cases this is treated as a soft constraint (Cheong et al. 2009, Sabar, Ayob, Qu & Kendall 2012, Amaral & Pais 2016, Burke & Bykov 2017, Leite et al. 2019). Other common objectives include scheduling all the exams in as few sessions as possible (Caramia et al. 2008, Cheong et al. 2009, Cavdur & Kose 2016), maximising spacing between exams (Eley 2007, Burke et al. 2007, Caramia et al. 2008, Qu et al. 2009, Kahar & Kendall 2010, Sabar, Ayob, Qu & Kendall 2012, Li et al. 2012, Kahar & Kendall 2015, Cavdur & Kose 2016, Woumans et al. 2016, Burke & Bykov 2017, Leite et al. 2019), scheduling higher attendance exams earlier in the schedule (Burke et al. 2007, Cheong et al. 2009, Sabar et al. 2009, Sabar, Ayob, Qu & Kendall 2012, Li et al. 2012, Burke & Bykov 2017, Leite et al. 2019), scheduling exams or groups of exams in preferred time slots and avoid some time slots (Cheong et al. 2009, Wang et al. 2010, Sabar, Ayob, Qu & Kendall 2012, Burke & Bykov 2017, Leite et al. 2019) and minimising the necessary rooms for an exam and the distance between them (Kahar & Kendall 2010, 2015).

Due to only having 6 days to schedule 250 exams for 4000 students the case of the United States Military Academy/West Point (Wang et al. 2010) becomes slightly different from the rest, as if treated as a regular examination scheduling problem it would often be infeasible. To solve this, the constraint "All the exams are scheduled once" had to be treated as a soft constraint and makeup exams have to be scheduled for some courses while trying to minimise the number of students who must take them and considering that some courses should not have make up exams. Woumans et al. (2016) introduces these constraints as well, not because of infeasibility problems, but as a way to create better solutions regarding the spacing between exams.

There are some similarities between this problem and the thesis defence scheduling problem. Due to both being academic scheduling problems, the resources are similar in both. For example, the number of rooms, professor availability, number of professors, amongst others. Thus, the hard constraints regarding these resources should be similar for both problems. Furthermore, as scheduling problems where each event (exam or thesis defence) must be scheduled, the constraints that ensure this happens are also usually alike. In the Thesis Defence Scheduling literature all the defences are scheduled in predefined sessions, however, in our case they can start at any hour, which is also the case in the Exam Scheduling literature.

Whereas in exam scheduling the goal is to provide more time for students to prepare for their exams, in thesis defence scheduling the objective is to create more compact schedules for the committees. This means that the most common objective in exam scheduling, maximising the spacing between exams, is actually the opposite of one of the thesis scheduling goals at IST.

A summary of the literature analysed in this section is presented in Table 3.2.

Table 3.2: Summary of the Literature on Exam Scheduling Problems

		Objective							
Paper	**Solution Method**	1)	2)	3)	4)	5)	6)	7)	8)
Eley (2007)	Ant Colony		✓						
Burke et al. (2007)	Hyper-Heuristic		✓	✓					✓
Caramia et al. (2008)	Local Search	✓	✓						
Cheong et al. (2009)	Evolutionary Algorithm	✓		✓	✓		✓		
Qu et al. (2009)	Hybrid Heuristic		✓						
Sabar et al. (2009)	Graph Colouring			✓					
Wang et al. (2010)	Binary Programming				✓	✓			
Kahar & Kendall (2010)	Constructive Heuristic		✓					✓	
Sabar, Ayob, Qu & Kendall (2012)	Hyper-Heuristic		✓	✓	✓		✓		✓
Li et al. (2012)	Pattern Recognition		✓	✓					✓
Elloumi et al. (2014)	Variable Neighbourhood								✓
Kahar & Kendall (2015)	Great Deluge		✓					✓	
Cavdur & Kose (2016)	Binary Programming	✓	✓						✓
Woumans et al. (2016)	Column Generation		✓			✓			
Amaral & Pais (2016)	Tabu Search						✓		
Burke & Bykov (2017)	Hill Climbing		✓	✓	✓		✓		✓
Leite et al. (2019)	Simulated Annealing		✓	✓	✓		✓		✓

1)Schedule all the exams in as few sessions as possible; 2)Maximising spacing between exams creating more even schedules; 3)Scheduling higher attendance exams earlier in the schedule; 4)Scheduling exams or groups of exams in preferred time slots and avoid some time slots; 5)Minimising extra exams; 6)Minimising exams taken consecutively in the same and/or different days;7)Minimise the necessary rooms for an exam and the distance between them; 8)Other.

3.3 Course Timetabling

Course Timetabling is the other original branch in academic scheduling, thus, similarly to exams scheduling, there is extensive literature in the area. Therefore, only a portion will be analysed. All the articles were searched in Web of Science, under the search topic "Course Timetabling" and category "Operations Research and Management Science" where then the 25 most cited papers from 2007 until 2014 and the 8 most cited papers from 2015 to the present day, excluding literature reviews or unrelated articles and that were available through the IST VPN service, were selected.

A variety of approaches have been applied to this problem, such as integer programming (Schimmelpfeng & Helber 2007, Lach & Luebbecke 2012, Phillips et al. 2015, Vermuyten et al. 2016), with the latter using a two-step integer programming approach to better solver real world instances, simulated annealing (Ceschia et al. 2012, Bellio et al. 2016), tabu search (Lü & Hao 2010), harmony search (Al-Betar & Khader 2012), stochastic optimization (Pongcharoen et al. 2008), graph colouring (Burke et al. 2010), ant colony optimization (Nothegger et al. 2012), local search (Müller & Rudová 2016, Goh et al.

2017), honey bee mating (Sabar, Ayob, Kendall & Qu 2012) evolutionary algorithm (Beligiannis et al. 2008), MaxSat (Achá & Nieuwenhuis 2014), hyper and meta heuristics(Burke et al. 2007, Qu & Burke 2009, De Causmaecker et al. 2009, Bai et al. 2012, Soria-Alcaraz et al. 2014, Lewis & Thompson 2015, Soria-Alcaraz et al. 2017) and genetic algorithms (Yang & Jat 2011, Akkan & Gulcu 2018).

There is a more noticeable variance in hard constraints for this problem in comparison with exams scheduling. The reason for this is mainly related to different universities having different rules, guidelines, students and resources. The following rules are the most common hard constraints found in the literature:

• Every class must be scheduled exactly once;

• One room can only hold one class at a time;

• Students must be able to attend all their classes;

• One professor cannot teach two classes at the same time;

• Classes can only be assigned to professor of the same field;

• The required number of professors must be met for each class;

• Days or times of the day where there must not be classes;

• Room where classes are taught must fulfil requirements such as seating capacity, projector, etc.

There are some cases where these general hard constraints cannot be fulfilled. Thus, some authors adapt and treat them as soft constraints. For instance, in some cases, there might not be enough rooms with the necessary seating capacity for some classes. Hence, minimising the students that would go unseated in a full attendance scenario becomes the objective (Schimmelpfeng & Helber 2007, De Causmaecker et al. 2009, Burke et al. 2010, Lü & Hao 2010, Lach & Luebbecke 2012, Achá & Nieuwenhuis 2014, Bellio et al. 2016). In addition, professor requests regarding rooms, consecutive classes or teaching periods can be treated as a hard constraint, however this is often regarded as one of the soft constraints (Schimmelpfeng & Helber 2007, Pongcharoen et al. 2008, Phillips et al. 2015, Vermuyten et al. 2016). Another concern regarding the staff is to regulate the hours they must teach each week or day (Pongcharoen et al. 2008, Beligiannis et al. 2008).

An inquiry to each student for his/hers individual preferences would be more complicated to include in the model. The common approach is to consider general preferences as a means to create a superior schedule for the students. This creates a number of objectives, such as creating more compact timetables by reducing idle periods, which can also be applied to the staff (Beligiannis et al. 2008, Burke et al. 2010, Lü & Hao 2010, Lach & Luebbecke 2012). This goal is treated as a hard constraint by Vermuyten et al. (2016). Another objective is to reduce changes in classrooms and travel times for students, as this creates an inconvenient and often times congestion in the hallways (Pongcharoen et al. 2008, Phillips et al. 2015, Vermuyten et al. 2016).

An important concern in the literature is that the proposed models can be compared with each other in terms of computation times and quality of the solution, thus many authors opt to solve the same problem and use the same instances in order to have a better benchmark on the quality of their proposals. Frequently, the soft constraints are similar to the one used in the ITC2007 competition tracks (Lewis et al. 2007, Di Gaspero et al. 2007). The track 2 (Burke et al. 2007, Qu & Burke 2009, Yang & Jat

2011, Al-Betar & Khader 2012, Sabar, Ayob, Kendall & Qu 2012, Ceschia et al. 2012, Bai et al. 2012, Soria-Alcaraz et al. 2014, Lewis & Thompson 2015, Soria-Alcaraz et al. 2014, Goh et al. 2017, Akkan & Gulcu 2018) is about post enrolment based course timetabling and encompasses the following soft constraints: reducing days where a student only has one class and freeing up the last time slot of the day and preventing students from having more than two classes in a row. The track 3 (Burke et al. 2010)is on curriculum based course timetabling and deals with the following soft constraints: minimising the students that would go unseated, reducing idle periods and regulating the spread of lectures of the same course. Due to the variety of hard constraints present in this problem, there are a few which are

Table 3.3: Summary of the Literature on Course Timetabling Problems

		Objective							
Paper	**Solution Method**	1)	2)	3)	4)	5)	6)	7)	8)
Schimmelpfeng & Helber (2007)	Integer Programming	✓	✓						✓
Burke et al. (2007)	Hyper-Heuristic				✓				
Beligiannis et al. (2008)	Evolutionary Computation			✓			✓		✓
Pongcharoen et al. (2008)	Stochastic Optimization		✓					✓	✓
Qu & Burke (2009)	Hyper-Heuristic				✓				
De Causmaecker et al. (2009)	Metaheuristic	✓							✓
Burke et al. (2010)	Graph Colouring	✓		✓			✓		
Lü & Hao (2010)	Tabu Search	✓		✓		✓	✓		
Jat & Yang (2011)	Genetic Algorithm				✓				
Lach & Luebbecke (2012)	Integer Programming	✓		✓		✓	✓		
Al-Betar & Khader (2012)	Harmony Search				✓				
Sabar, Ayob, Kendall & Qu (2012)	Honey-Bee Mating				✓				
Ceschia et al. (2012)	Simulated Annealing				✓				
Nothegger et al. (2012)	Ant Colony				✓				
Bai et al. (2012)	Hyper Heuristic				✓				
Achá & Nieuwenhuis (2014)	MaxSat	✓				✓	✓		✓
Soria-Alcaraz et al. (2014)	Hyper-Heuristic				✓				
Lewis & Thompson (2015)	Hyper-Heuristic				✓				
Phillips et al. (2015)	Integer Programming		✓			✓		✓	✓
Bellio et al. (2016)	Simulated Annealing	✓		✓		✓	✓		
Vermuyten et al. (2016)	Integer Programming		✓					✓	✓
Müller & Rudová (2016)	Local Search								✓
Soria-Alcaraz et al. (2017)	Hyper Heuristic				✓				
Goh et al. (2017)	Improved Local Search				✓				
Akkan & Gulcu (2018)	Genetic Algorithm				✓				

1)Minimising the students that would go unseated; 2)Maximising satisfaction of professor requests; 3)Reducing idle periods; 4)ITC 2007 competition track 2; 5)Minimising the number of classrooms used for the same course; 6)Regulating the spread of lectures from the same course; 7)Reducing travel times and room changes; 8)Other.

not adaptable to the master's thesis defence problem. Nonetheless, as it was the case for the exam scheduling problem, the resources are similar between all the academic scheduling problems, thus, the constraints that limit the use of these resources are similar to an extent.

The main goal of this field is to improve the quality of the schedule both for students and professors, and while the individual objectives that are present in the literature might not be the same as the ones that exist in thesis defence scheduling, there are some similarities. For instance, in course timetabling there are different parties, namely the students and the professors, who have different and sometimes conflicting preferences, similarly to the thesis defence problems where different examination committee members have different necessities and some aspects on how this was handled can be adapted to our problem. Another characteristic of this field is that sometimes instead of defining what the goals are, the professors are asked individually what are their preferences and then those are included in the model. This can be adapted to the thesis defence scheduling problem, where often different examination committee members have different requests, mainly regarding the days when they are available to be at a thesis defence.

A summary of the literature analysed in this section is provided in Table 6.1.

3.4 Preference Modelling

Preference modelling is an important aspect to consider not only in operations research but also in a variety of other fields, such as economy, sociology, psychology, political science, artificial intelligence and decision-making analysis (Öztürké et al. 2005). For the purpose of this thesis, the analysis will be focused primarily on how this tool is used in operations research related problems. One of the difficulties of representing user preferences is that it is harder to assign values to them than to other traditional constraints (Meseguer et al. 2006). Thus, it is important to understand how this usually done in the literature.

Just like exams and timetable scheduling, the primary goal in the thesis defence scheduling problem is to include the preferences of professors and students in order to create a more desirable schedule. Thus far, the main focus in the literature is to create general rules that represent preferences that are shared by most of the staff and using those as the components of the objective function. Nonetheless, in the case of IST it is important to differentiate the preferences of the professors as they are often different, thus, it is important that our model allows for their inclusion in addition to the more general rules, as is done in course timetabling literature where there are some examples where individual requests are introduced in the models (Schimmelpfeng & Helber 2007, Pongcharoen et al. 2008, Phillips et al. 2015, Vermuyten et al. 2016).

While there were some works in the academic scheduling literature where these individual preferences are taken into account, this is not widespread in the literature, where general rules for creating better quality solutions are the norm. Therefore, it is necessary to review additional literature on how this is done in other scheduling problems. All the additional articles in this section were searched in Web of Science, under the search topics "Scheduling" and "Preferences" and category "Operations Research and Management Science". The six most cited papers from the last 5 years, excluding literature reviews or unrelated articles and that were available through the IST VPN service, were selected.

All the new papers were from areas related to healthcare scheduling, such as home health care problems where the goal is to optimise the operational costs of the companies (Braekers et al. 2016, Shi et al. 2017, Zhang et al. 2019), patient admission problems which aim to assign patients to hospital beds (Turhan & Bilgen 2017), nurse rostering problems with the objective of allotting shifts to nurses (Rahimian et al. 2017) and operating theatre timetabling problems which seek to schedule surgeries Penn et al. (2017). There are several solution methods to solve these problems, nonetheless, integer programming based methods are the most common one (Schimmelpfeng & Helber 2007, Phillips et al. 2015, Vermuyten et al. 2016, Rahimian et al. 2017, Penn et al. 2017) while stochastic optimization (Pongcharoen et al. 2008), hybrid genetic algorithms (Shi et al. 2017), fix and relax and fix and optimise (Turhan & Bilgen 2017), meta heuristic (Braekers et al. 2016) and simulated annealing (Zhang et al. 2019) are being used as well.

Most papers opt for an approach where the inconvenience to users is minimised (Schimmelpfeng & Helber 2007, Pongcharoen et al. 2008, Vermuyten et al. 2016, Braekers et al. 2016, Shi et al. 2017, Turhan & Bilgen 2017, Rahimian et al. 2017), whereas there are some cases where the choice is to maximise the benefits (Phillips et al. 2015, Penn et al. 2017, Zhang et al. 2019). Besides individual preferences, all papers consider other general rules or cost functions. By definition, all of these are naturally scaled differently, therefore, methods to compare these different objectives are usually implemented. It is common practice to use weights for all preferences that, can not only be useful for scaling them in a way that turns them comparable to other objectives, but are also useful as they allow users to set which soft constraints they deem to be the most important relative to one another, by assigning them different weights. There are some cases where not only different soft constraints are given different weights, but there are also weighted distinctions between different groups of people (Vermuyten et al. 2016, Zhang et al. 2019). A different model is presented in Braekers et al. (2016), which introduces two different objectives. The first objective regards cost reduction and the second inconvenience to the patients, that is, it evaluates the compliance with their stated preferences by factoring in the assigned health care worker and the advance or delay regarding the preferred time window, considering whether this time window exists within the feasible solutions. Consequently, there is no need to assign weights to scale the objectives, instead a set of efficient solutions is presented, and the user can determine which one they deem the preferred one. The authors state that it is hard to define weights *a priori* and chose to study the trade-offs between the different objectives. This is a widely shared belief within the literature, which more often than not leaves the decision of assigning weights up to the future users of their work.

The most common requests included in the models are time related (Schimmelpfeng & Helber 2007, Pongcharoen et al. 2008, Vermuyten et al. 2016, Braekers et al. 2016, Rahimian et al. 2017, Penn et al. 2017, Zhang et al. 2019), going from preferred time slots for certain events, to preferred shifts or days off, which is expected due to the scheduling nature of all problems. Nonetheless, other types of requests, such as preference over nurses, types of bed or teaching formats are also accepted by some models (Schimmelpfeng & Helber 2007, Pongcharoen et al. 2008, Phillips et al. 2015, Braekers et al. 2016, Turhan & Bilgen 2017). In order to represent different preferences, the people for whom the schedules are aimed can be given the choice between several preference levels that will influence the weight of each scheduled event (Schimmelpfeng & Helber 2007, Phillips et al. 2015, Braekers et al. 2016, Penn et al. 2017, Zhang et al. 2019). In models where the objective function is being maximised, the lower preference score values represent the least preferred options whereas the higher values represent the most desirable options, with the opposite being true for minimisation based models. However, it is more common that preferences are given binary values (Schimmelpfeng & Helber 2007, Pongcharoen et al. 2008, Vermuyten et al. 2016, Turhan & Bilgen 2017, Rahimian et al. 2017), where, for maximisation problems, 0 represents non-compliance with the request and 1 represents compliance and vice-versa for minimisation problems. The reasoning between these two different methods is that in some cases a request can either be complied with or not, with no options in between, for example, someone can be scheduled for a day-off or not, whereas in other cases there is a different level of desirability between more than two options, for example, if patients request that they are assisted in a set time period, a delay or advance of 30 minutes might not have the same undesirability as a one-hour delay.

Some models also include mechanisms to differentiate between different requests or different scheduling situations. Schimmelpfeng & Helber (2007) introduces a way to penalise some requests that are

Table 3.4: Summary of the Literature on Preference Modelling

	Topic		**Solution Method**		**Individual Preferences**		**Preference Modelling Method**					
Paper	CT	HC	IP	O	TB	O	1)	2)	3)	4)	5)	6)
Schimmelpfeng & Helber (2007)	✓		✓		✓	✓	✓		✓	✓		✓
Pongcharoen et al. (2008)	✓			✓	✓	✓	✓		✓			
Phillips et al. (2015)	✓		✓			✓		✓		✓		
Vermuyten et al. (2016)	✓		✓		✓		✓		✓		✓	
Braekers et al. (2016)		✓		✓	✓	✓	✓			✓		✓
Shi et al. (2017)		✓		✓		✓	✓					
Turhan & Bilgen (2017)		✓		✓		✓	✓		✓			
Rahimian et al. (2017)		✓	✓		✓		✓		✓			
Penn et al. (2017)		✓	✓		✓			✓		✓		✓
Zhang et al. (2019)		✓		✓	✓			✓		✓	✓	

Topic: CT) Course Timetabling; HC) Healthcare

Solution Method: IP) Integer Programming based methods; O) Other

Individual Preferences: TB) Time based; O) Other

Preference Modelling Method: 1) Minimise inconvenience regarding general rules and/or individual preferences; 2) Maximise benefits regarding general rules and/or individual preferences; 3) Binary Preferences; 4) Preference degrees; 5) Distinction between different groups in individual preferences; 6) Increased penalization for infeasible/detrimental preferences or for multiple non-compliance scenarios with the same person's preferences deemed detrimental to other users, specifically professors requesting rooms that might not have enough seating capacity for all the students, Braekers et al. (2016) devalues requests that are nearly unfeasible and Penn et al. (2017) creates an incremental penalty for instances where staff has several surgeries scheduled for less preferred time slots.

Similarly to most cases in this section, the thesis defence scheduling case at IST combines general rules and individual requests, thus, the structure that most papers use to define their soft constraints can be translated to the case at hand. Furthermore, there were some interesting concepts that might be of use, such as distinction between different staff regarding the positions they occupy (Vermuyten et al. 2016, Zhang et al. 2019), incremental penalties for cases where one user is assigned several least preferred time slots (Penn et al. 2017) or the penalization of requests that are deemed detrimental to other users if they were to be complied with (Schimmelpfeng & Helber 2007).

A summary of the literature analysed in this section is provided in Table 3.4.

3.5 Scheduling for Fairness

In scheduling problems, there is often a trade-off between finding the best overall solution and guaranteeing that this solution is fair for all the affected. Furthermore, fairness is a subjective concept, which often varies from field to field and sometimes even within the same field. This section is focused on the fairness goals and methods taken to achieve those goals.

Besides the literature that has been introduced up to this point, the three most cited papers from the last ten years, searched on Web of Science under the search topics "Scheduling" and "Fairness" and category "Operations Research and Management Science", that were not reviews or unrelated, were selected. The themes of these new articles are healthcare (Stolletz & Brunner 2012), traffic flow management in an airport (Barnhart et al. 2012) and scheduling of earth observations made by a satellite (Tangpattanakul et al. 2015).

Two different groups of fairness goals can be identified in the literature. The intent of the first one is to create models in a way that is as respectful to the right to equality of all the affected as possible, be it by evening out workloads (Stolletz & Brunner 2012, Kochaniková & Rudová 2013, Dung et al. 2015, Battistutta et al. 2019), the spread of events (Sabar, Ayob, Qu & Kendall 2012, Burke & Bykov 2017, Leite et al. 2019), the assignment of undesirable time slots (Penn et al. 2017) or the profit each individual can take from the scheduling (Tangpattanakul et al. 2015). The aim of the second group is to differentiate each affected by the scheduling based on certain characteristics, such as order in which a request was made, in a first-come first served basis (Barnhart et al. 2012), position of the professional (Vermuyten et al. 2016) or the number of predetermined events someone must attend (Fastré 2017).

There is an interesting dichotomy between these two groups that alludes to the equity *versus* equality debate. Nonetheless, in this particular setting, there is a clear distinction between the cases where each group is implemented, meaning that there is no universal solution. The first one is being used in cases where each affected has very similar characteristics and thus, by equalising the outcomes of their schedules, a fairer result is reached. In contrast, there are cases where striving to distinguish different actors can be viewed as the fairest approach, either because sometimes equalising results is not possible, for example, a scheduling problem where everyone could prefer to be served first, means that distinctions must be made to find out who is more deserving of the preferred option (Barnhart et al. 2012) or because different affected have different enough starting characteristics that a distinction must be made to ensure that they are treated fairly (Vermuyten et al. 2016, Fastré 2017).

A number of different methods are used in the literature to ensure that the aforementioned fairness goals are achieved. Some of them, by definition, are more suitable to one group of goals than the other.

The first method is based on minimising the difference between the best and the worst assigned individual schedules (Stolletz & Brunner 2012, Dung et al. 2015, Tangpattanakul et al. 2015). It is very important when you are using this method to properly define the weights for the objective function, as if you assign too much weight to this constraint there might be cases where it becomes optimal to worsen the scheduling for someone without actually improving the situation of someone else, and while you could argue for the fairness of that premise, it is usually not the envisioned result of the model. This method is often modelled in a multi-objective problem framework or similar, which means that it will be up to the user to select one of the efficient solutions, giving the possibility to not choose one of the problematic solutions presented above. This method was used exclusively in the first fairness goal group, as it forces an equalised solution for all the individual schedules.

The second method is based on creating incremental penalties for multiple occurrences of certain events for the same individual. This guarantees that a certain actor does not end up having multiple undesirable occurrences in detriment of others having their preferred option. This method is more flexible than the previous one, as it can be used in both the first (Penn et al. 2017, Battistutta et al. 2019) and second group of fairness goals (Barnhart et al. 2012). There are different ways of modelling this method, for example, the increments can be quadratic (Barnhart et al. 2012, Battistutta et al. 2019) and give a higher penalty for each occurrence or linear (Penn et al. 2017) and give similar penalties to each new occurrence.

The third method is flexible as well, being used in problems of the first (Sabar, Ayob, Qu & Kendall 2012, Burke & Bykov 2017, Leite et al. 2019) and second (Barnhart et al. 2012) group of fairness goals. This method consists of creating a metric, based on the characteristics of each problem, that can be maximised or minimised in the objective function and allows to measure the fairness of the schedules. This is probably the most flexible method, as the metric can be formulated in whichever way the problem asks for. This was presented as the common practice in the industry of traffic flow management, but it was estimated that the second method, explained in the previous paragraph, could be adapted to this problem more efficiently with minimum losses in solution quality (Barnhart et al. 2012).

The next method is exclusive to problems of the second fairness goal group (Vermuyten et al. 2016, Fastré 2017), as it distinguishes between different actors by assigning them different weights in the objective function based on their predetermined characteristics. There can be a plethora of different deciding factors when assigning these weights, of which the following can be highlighted: different position within a company or institution (Vermuyten et al. 2016) or different number of predetermined mandatory presences in certain events (Fastré 2017).

The last method guarantees fairness by forcing fair solutions with a constructive heuristic, and then not allowing them to be less fair in the improvement step (Kochaniková & Rudová 2013). This method was only used in problems of the first fairness goal group, but it might be used in the other group with some adjustments.

In the thesis defence scheduling problem at IST, cases where both fairness groups can be applied are found. Firstly, there is a clear distinction between the groups that compose the examination committee, that is, the chairperson, supervisor and examiner as well as a distinction between the number of thesis each of the professors must attend, alluding to a second fairness goals group. Nevertheless, there are some members within each of the aforementioned groups that have similar starting characteristics, which might entail some consideration regarding the usefulness of the first fairness group goals and methods in this case as well.

A summary of the literature analysed in this section is provided in Table 3.5.

Table 3.5: Summary of the Literature on Scheduling for Fairness

			Fairness						
	Topic		**Goal**		**Method**				
Paper	AS	O	B	D	1)	2)	3)	4)	5)
Stolletz & Brunner (2012)		✓	✓		✓				
Barnhart et al. (2012)		✓		✓		✓	✓		
Sabar, Ayob, Qu & Kendall (2012)	✓		✓				✓		
Kochaniková & Rudová (2013)	✓		✓						✓
Dung et al. (2015)	✓		✓		✓				
Tangpattanakul et al. (2015)		✓	✓		✓				
Vermuyten et al. (2016)	✓			✓				✓	
Fastré (2017)	✓			✓				✓	
Penn et al. (2017)		✓	✓			✓			
Burke & Bykov (2017)	✓		✓				✓		
Battistutta et al. (2019)	✓		✓			✓			
Leite et al. (2019)	✓		✓				✓		

Topic: AS) Academic Scheduling; O) Other

Fairness Goal: B) Balance and even out the schedules, considering that every person has an equal standing and right to equality; D) Differentiate the importance level given in the model to people in different relative positions

Fairness Method: 1) Minimise the maximum difference between the best assigned individual schedule and the worst; 2) Incremental penalty for multiple occurrences of certain events; 3) Creation and incorporation in the objective function of a metric that measures fairness; 4) Different weight assignments for people in different positions; 5) Constructive heuristic guarantees fairness

3.6 Chapter Considerations

Besides thesis defence scheduling, academic scheduling has two other main branches, exams scheduling and course timetabling. In this review, an analysis of the characteristics of these problems was made, more specifically on the adopted solution methods, hard constraints and objectives, with a focus on the thesis scheduling problem. This research can be found on sections 3.1, 3.2 and 3.3, respectively. Then, in the following two sections, 3.4 and 3.5, further insight is given on some elements of these problems that were deemed important for the case at case, namely preference modelling and scheduling for fairness, respectively, with a deeper review of some articles presented up until that point and some new ones of different areas as well.

There were a few solution methods applied to the thesis defence scheduling problem, nonetheless, the authors used different cases, which makes comparing them difficult. Battistutta et al. (2019) makes a comparison between different methods, deeming mixed integer programming the best one for their instances. There is no case where the entirety of the examination committee compositions is already defined before the model assigns them to time slots in contrast with the case at IST, there is also no case where there are no predefined time slots where the defences can be assigned, which is different from the case at hand, where they may start at any hour. There were a few different objectives in the literature, with some of them being related to balancing out workloads and assigning suitable examination committee compositions to defences, which does not apply in our case. Furthermore, there is no focus on increasing the flexibility of individual examination committee members preferences within the literature.

The other fields in academic scheduling have been more explored, which translates in a higher variety of solution methods being used, as well as an emphasis on using the same instances in order to have better means of comparison. There is also a plethora of objectives, with some of them being translatable to the thesis defence scheduling problem. The course timetabling literature proved to have more applicable concepts for the specific case approached in this work, as examples of concepts such as individual requests or reducing idle periods. Other aspects such as minimising room changes and travel times, which are found in both course timetabling and exam scheduling, could also be applicable. However, they are out of the scope of this work, as assigning the rooms is the responsibility of a different service. This disparity in translatability between the three problems can be explained by the fact that in exam scheduling the main goal is to provide students the appropriate time to prepare, whereas in course timetabling and thesis defence scheduling it is more common that the affected prefer to have more compact schedules.

In preference modelling there were different types of preferences being represented, with the majority being time related and there was also a distinction between articles that would distinguish the individual preferences of different groups of people, with the consideration that they might have different weights in the scheduling problems at hand. There was also the decision between representing preferences as a binary, where they are either respected or not or representing them with multiple levels, meaning that there would be several satisfaction levels for the same preference.

There were two different groups that were identified in scheduling for fairness. Some papers focus more on guaranteeing that all the affected by the scheduling are given the most equal outcomes as possible, whereas some other papers consider that not all the actors have the same characteristics and thus, they make a distinction in the weight of each actor based on the characteristics deemed necessary. Furthermore, several methods to ensure that this is achieved were found, with some of them being used exclusively for one of the two aforementioned groups while others can be used for both.

Chapter 4

Mathematical Model

In this chapter, the problem of scheduling the thesis defences is formalised for the case at hand. In Section 4.1 the problem is described. Afterwards, in Section 4.2, the proposed formulation is introduced, including its sets, parameters and variables, and its several groups of constraints, namely the structural integrity constraints (Subsection 4.2.1), the Taguspark *Campus* presence constraints (Subsection 4.2.2) and the compactness constraints (Subsection 4.2.3). Finally, the four objectives are addressed in Subsection 4.2.4 and a conclusion to the chapter, including parallels with the literature, is made in Section 4.3.

4.1 Problem Description

The main goal of the Thesis Defence Scheduling problem is to find a schedule that assigns each defence (or as many as possible) to a given time slot while respecting some constraints, such as the availability of committee members or rooms.

As previously stated, the examination committees for the MEGI defences are composed by three members, who must all be available at the time the defence is scheduled. They are the chairpersons, who are usually the ones who must attend the most defences, the supervisors, who have guided the defendants through their dissertation and an additional member. Furthermore, in the literature, there are some considerations that are not applicable to the MEGI case. For instance, it is often the decision-maker's responsibility to not only schedule the defences, but to entirely or partially assign the committee for each one. Nonetheless, for the case at hand, that assignment is done prior to the scheduling. Additionally, there are also cases in the literature where the room assignment is the responsibility of the decision-maker that schedules the defences, which, again, is not the case for the MEGI discussions, as that is done by a different service. However, there is still a limited number of rooms, which means that it is necessary to not schedule more defences than there are available rooms for each time slot.

Since MEGI defences are held in the Taguspark *Campus*, often committee members have to travel there solely for the purpose of being present for a dissertation discussion. For this reason, one of the most pressing issues while scheduling the master's thesis defences is to guarantee that each member has defences scheduled in as few days as possible. Furthermore, it is also a frequent occurrence that committee members might have different preference levels for different days or times and, although being available at a certain time if that is necessary, they would rather be scheduled for a different one. Additionally, they also usually prefer to have all their defences in a row, having as much of a compact schedule as possible.

Thus, four different objectives that affect the quality of a proposed schedule can be identified. Firstly, there is the objective of scheduling as many defences as possible, which is commonly referred to as a hard constraint in the literature, in cases where it is known, prior to the scheduling process, that all discussions can be scheduled. Moreover, it is also paramount that the number of days a member is scheduled for is minimised, the individual preferences for time slots are taken into account and that the schedules are as compact as possible.

4.2 Problem Formulation

In this section, all the data that serves as input for the mathematical model, which is a Mixed Integer Linear Programming (MILP) model, is presented, including sets and parameters. Furthermore, both the main decision and auxiliary decision variables will be addressed as well as the several groups of constraints and, lastly, its objectives.

Let T denote the set of thesis defences that need to be scheduled. These thesis must be scheduled inside a defined set of days D. Moreover, within each day, only certain times, regarded as H, are available for a defence to start at. For the purpose of this work, time is divided into slots of 15 minutes, that is, if hour 0 is the first available time for a defence to start, hour 1 is the time 15 minutes after hour 0, as that is how the thesis defences are currently scheduled, that is they can start at minutes 0, 15, 30 and 45 of any given hour. Each defence has already been assigned three committee members. They can be either its chairperson, the supervisor or an additional member. This set of three positions is represented as P. Moreover, the set of all committee members is denoted as M. The parameter that represents

A necessity to differentiate some members from the others in terms of their importance for scheduling purposes was identified, which is achieved by considering different weights to be applied. Specifically, there is the case of the chairpersons, who have a higher average number of defences to attend and for which the weight is defined as MW. Furthermore, there might be other reasons for assigning different scheduling weights to committee members, such as members who do not teach or often travel to the Taguspark *Campus*, therefore, a second weight O, to distinguish between members was added. Allowing the decision-maker to instead set the weight for each individual member was considered, but that would lead to many other fairness considerations as well as a more arduous process, thus, only these two categories to distinguish between members in different positions were added.

Moreover, the possibility of assigning different preference levels each member has for certain time-slots is also regarded in the model, with the addition of the parameter HP. This parameter assigns, for each committee member and time-slot, an integer number representing their preference, with larger numbers representing larger preference levels. Furthermore, to represent unavailability, the assigned level should be 0.

As the main purpose of this problem is to schedule thesis defences, in other words, assigning them to time-slots, the main decision variables represent this process. Thus, let $X_{t_{d_h}}$ be a binary variable that takes the value 1 if thesis defence $t \in T$ is scheduled for day $d \in D$ and to start at hour h, and 0 otherwise.

Furthermore, to assign values to some objective functions, two groups of auxiliary variables were added. Firstly, there is the group that will aid in minimising the number of days a member has discussions scheduled on, comprised by the integer variable G_m and the binary variables Y_{md} and GQ_{mq}. Note that the set Q, representing the possible options for numbers of days jury members have defences scheduled on, was added to aid in the definition of these variables as well as another parameter B, which represents the maximum number of days a member can have a thesis scheduled on. This will be further explained in Subsection 4.2.2.

Lastly, there is the group of auxiliary variables intended to measure the compactness of schedules, which is composed by the integer variables C_{tdh} and Z_{tdh}. Observe that the set CZ corresponds to the zone, divided by quarter-hours, after a member ends a dissertation discussion, where scheduling a new defence would be considered compact. Furthermore, each of those quarter hours has a certain weight CZW_{cz}. This will be further explained in Subsection 4.2.3.

4.2.1 Structural Integrity Constraints

There are four different structural integrity constraints. Firstly, there is the constraint that guarantees that a defence cannot be scheduled more than once (4.1). Secondly, there are two constraints that ensure the availability of committee members for the time-slots they are scheduled at. This is achieved by making sure that no defence they are scheduled for occurs at a time when they have stated to be unavailable (4.2). To ensure this happens, the aforementioned constraint states that the decision variable can never be greater than the preference level HP for the committee members of the corresponding thesis. Thus, a preference level can never be a non-integer number lower than 1 and if the committee member is unavailable, this value must be 0.

Furthermore, it is also necessary to guarantee that committee members have no juxtaposed defences (4.3). To ensure this, the constraint states that if two different thesis defences have committee members in common, one of them cannot have been scheduled to start at either the same time or up to L - 1 quarter-hours before the other one is to start, with L being the duration of a full dissertation discussion. Finally, there is a constraint that ensures that there are no more thesis defences at a time than rooms available (4.4). This constraint follows a similar logic to the previous one, but without the necessity of checking whether the thesis defences in question have committee members in common. Thus, the reasoning behind it is that at most, in every time period from h-(L-1) to h there can never be a larger number of thesis defences starting than there are rooms available. It is important to note that for the last two constraints, (4.3) and (4.4), it is not sufficient to consider only one h at a time instead of having a sum from h-(L-1) to h. This is the case due to the decision variable $X_{tdh} \in \{0, 1\}$ representing

Table 4.1: Sets, Parameters and Variables

Sets and Indexes

$t, t_1 \in T$	Set of master thesis defences
$m \in M$	Set of committee members
$d \in D$	Set of the available days
$h \in H$	Set of available hours
$p \in P$	Set of possible positions a committee member may occupy
$q \in Q$	Set of possible numbers of days jury members have defences scheduled on
$cz \in CZ$	Set of quarter hours designating the compactness zone

Parameters

L	Length of a thesis defence in quarter hours
R	Number of available rooms
B	Maximum number of days a committee member can be scheduled for
BM	Big M
EXP	Exponential penalty for each additional day a member has a defence in
MWW	Weight for chairpersons
OW	Weight for members with other distinguishable factors
$CM_{t_p \subseteq M}$	Jury composition of each thesis defence
$ICM_{t,t1} \in \{0, 1\}$	1 if *t* and *t1* have any member in common, 0 otherwise
$HP_{mdh} \in \mathbb{N}_0$	Preference levels, 0 represents unavailability
$MW_m \in \{1, MWW\}$	MWW if member *m* is one of the chairpersons, 1 otherwise
$O_m \in \{1, OW\}$	OW if member *m* has a higher scheduling weight, 1 otherwise
$CZW_{cz} \in \mathbb{N}$	Weight for each quarter-hour in the compactness zone

Decision Variables

$X_{tdh} \in \{0, 1\}$	1 if thesis *t* is scheduled on day *d* to start at hour *h*, 0 otherwise

Auxiliary Variables

$Y_{md} \in \{0, 1\}$	1 if committee member *m* is part of a committee scheduled for day *d*
$G_m \in Q$	Number of days a committee member *m* has a defence in
$GQ_{mq} \in \{0, 1\}$	1 if committee member *m* has defences scheduled in *q* days
$C_{tdh} \in \{0, ..., BM\}$	Compactness impact of scheduling a defence t in time-slot d,h
$Z_{tdh} \in \{0, ..., BM\}$	Compactness impact of defence t scheduled in time-slot d,h

only a starting time for a thesis defence and not its whole duration.

$$\sum_{t=0}^{T} X_{tdh} \leq 1 \ \forall \ d \in D, h \in H \tag{4.1}$$

$$X_{tdh} \leq HP_{CM_{tp}dh} \ \forall \ t \in T, p \in P, d \in D, h \in H \tag{4.2}$$

$$X_{tdh} + \sum_{l=0}^{L-1} X_{t1,d\ ,h-l} \leq 1 \ \forall \ t \in T, t1 \in T, d \in D, h \in H, t \neq t1, ICM_{t,t1} = 1 \tag{4.3}$$

$$\sum_{t=0}^{T} \sum_{l=0}^{L-1} X_{t,d,h-l} \leq R \ \forall \ d \in D, h \in H \tag{4.4}$$

4.2.2 Taguspark *Campus* Presence Constraints

The Taguspark *Campus* presence constraints (4.5)–(4.10) set the values for variables Y_{md}, G_m and GQ_{mq}. In contrast to the previous group of constraints, these are not necessary to ensure the feasibility of the solutions. Instead, they are used to measure the number of days each member has a defence scheduled on. Thus, their goal is only to help evaluate the quality of the possible schedules. The particular strategy followed to set the values for these constraints had one primary goal, which was to add the possibility of having an exponential penalty in the objective function for each additional day every committee member has a thesis scheduled on. This ensures an additional level of fairness between members, as it will make it less likely that one of them ends up being heavily discriminated in favour of another by the model. Nonetheless, the model should keep its linearity as, otherwise, it can be too complex for a regular computer to solve efficiently.

The first two constraints, (4.5) and (4.6), set the value for the binary variable Y_{md}. The first of them ensures that in case there are no thesis defences with member m as part of their committee scheduled for day d, then Y_{md} is 0. On the contrary, the second constraint guarantees that if there is at least one discussion that committee member m is part of scheduled for day d, then Y_{md} is 1.

The former two constraints would be enough to include the objective of minimising the number of days a member has to be present for a defence, nonetheless, it still would not be possible to include the exponential penalty without compromising the linearity of the model. Thus, two other variables, G_m and GQ_{mq}, were added. The first is an integer variable that represents the number of days committee member m has discussions scheduled on (4.7), whereas the second is a binary variable that takes the value 1 if member m has thesis defences in q days, and 0 otherwise. With this binary variable, GQ_{mq}, it will be possible to include the exponential penalty in the objective function. Nonetheless, to be able to set its value, it is necessary to set an upper bound to the value of G_m (4.8) and include in the formulation the set Q, which is the set of possible number of days a member might have to be present for a dissertation discussion. To finish, the value of GQ_{mq} is set by two constraints, (4.9) and (4.10). The first one ensures that each member is assigned exactly one number of days to be present for a discussion, whereas the second finds out which number to assign each committee member.

$$\sum_{t1=0}^{T} \sum_{h=0}^{H} X_{t1,d,h} \geq Y_{CM_{tp}d} \ \forall \ t \in T, p \in P, d \in D, ICM_{t,t1} = 1 \tag{4.5}$$

$$X_{tdh} \leq Y_{CM_{tp}d} \;\; \forall \;\; t \in T, p \in P, d \in D, h \in H \tag{4.6}$$

$$\sum_{d=0}^{D} Y_{md} = G_m \;\; \forall \;\; m \in M \tag{4.7}$$

$$G_m \leq B \;\; \forall \;\; m \in M \tag{4.8}$$

$$\sum_{q=0}^{Q} GQ_{mq} = 1 \;\; \forall \;\; m \in M \tag{4.9}$$

$$\sum_{q=0}^{Q} GQ_{mq} \times q = G_m \;\; \forall \;\; m \in M \tag{4.10}$$

4.2.3 Compactness Constraints

The compactness constraints, (4.11)–(4.15), are intended to provide a measure of schedule compactness by setting the values for variables C_{tdh} and Z_{tdh}. Moreover, they are similar to the Taguspark *Campus* presence constraints, in the sense that they are only used to measure an objective function value and, thus, are not necessary to ensure feasibility.

Compactness, in the case at hand, evaluates how the proposed schedules perform in relation to the general preference of the committee members for having their defences sequentially. For this purpose, we define the set CZ, which corresponds to the number of quarter-hours after a defence has ended where it would still be considered acceptable to schedule another defence and still consider that schedule compact, which we will refer to as the compactness zone. For example, if $CZ = \{0, 1\}$ it would be considered compact if a member has another defence starting immediately after another one ends or if it would start a quarter-hour after that.

Thus, to measure this, we start by counting how many committee members of a certain thesis defence t have had another thesis defence t1 end within the compactness zone and, then, weigh their scheduling weight based on them being either a chairperson or being in another situation that warrants a different level of attention as well as the weight of the corresponding position on the compactness zone. This is done for every thesis and every time slot in constraint (4.11), with C_{tdh} being the integer variable that represents this measure. For example, lets say that $CZ = \{0\}$, considering a thesis defence t_1 with committee members 1, 2 and 3, every time another thesis defence t_2 ends, if it has members in common with t_1, then C_{tdh} takes the value of the number of members in common between t_1 and t_2. If there are multiple defences with members in common, then it becomes the sum of those.

Therefore, C_{tdh} measures the potential compactness impact of scheduling a thesis defence in a certain time slot. Nonetheless, to later evaluate how the schedule is performing regarding the objective, only the values of the variable for time slots that have been assigned to the referred thesis should be counted. Simply multiplying C_{tdh} for X_{tdh} would be mathematically correct, but it would compromise the linearity of the model. Thus, a big-M approach was taken, and implemented in constraints (4.12)–(4.15) with Z_{tdh}, which essentially represents the multiplication of C_{tdh} and X_{tdh}, taking the value of C_{tdh} if thesis t is scheduled on day d at hour h, and 0 otherwise.

Note that it is paramount that the big-M is as small as possible, preferably the upper bound to the corresponding variables. For the case at hand, this value can be found by multiplying MWW, OW, the biggest value for CZW and the number of different positions in the set P, as this is the biggest value C_{tdh} and, by extension Z_{tdh}, can take. Furthermore, it is important to take this into account when defining the weights MWW, OW and CZW, as, the larger they are, the larger the big-M is, which may impact the complexity when solving the model.

$$C_{tdh} = \sum_{t1=0}^{T} \sum_{m=0}^{M} \sum_{cz=0}^{CZ} MW_m \times O_m \times CZW_{cz} \times X_{t1,d,h-L-cz} \ \forall \ t \in T, d \in D,$$
$$h \wedge h - L - cz \in H, cz \in CZ, m \in CM_t \wedge CM_{t1}, t \neq t1 \quad (4.11)$$

$$Z_{tdh} \geq 0 \ \forall \ t \in T, d \in D, h \in H \quad (4.12)$$

$$Z_{tdh} \leq C_{tdh} \ \forall \ t \in T, d \in D, h \in H \quad (4.13)$$

$$Z_{tdh} \leq BM \times X_{tdh} \ \forall \ t \in T, d \in D, h \in H \quad (4.14)$$

$$Z_{tdh} \geq C_{tdh} - BM \times (1 - X_{tdh}) \ \forall \ t \in T, d \in D, h \in H \quad (4.15)$$

4.2.4 Objectives

For the case at hand, four different objectives to be maximised were considered. The first objective (4.16) regards the scheduling of the highest number of thesis defences possible. This sort of objective is more often than not approached as a hard constraint in the literature. Nonetheless, to be able to deal with possible data sets where some defences are not possible to schedule, it was regarded as an objective instead.

$$Max \sum_{t=0}^{T} \sum_{d=0}^{D} \sum_{h=0}^{H} X_{tdh} \quad (4.16)$$

The second objective (4.17) regards the maximisation of individual preferences for time slots, which can be, for every time slot, 0, if the member is unavailable at the time, or any other preference level the decision-maker chooses to implement, as long as they are integer and non-negative.

$$Max \sum_{t=0}^{T} \sum_{p=0}^{P} \sum_{d=0}^{D} \sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \quad (4.17)$$

The third objective (4.18) concerns the minimisation of the number of days members are scheduled to attend a defence, which, to facilitate the use of multi-objective approaches further on, we instead write as a maximisation objective. Furthermore, it introduces the possibility of including an exponential penalty for each additional day a member has a defence scheduled on, without compromising the linearity of the model. This improves fairness for each member, as it makes it less likely that the solution greatly benefits some members in this regard in the detriment of others, while excluding solutions that just create a worse schedule for one member, without improving the situation of another in the pursuit of fairness.

$$Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} \tag{4.18}$$

Lastly, as was previously explained, the fourth objective (4.19) regards the maximisation of the compactness of schedules. To achieve this, each thesis t is assigned a value Z depending on the time slot where it is scheduled. This value can go from 0, if none of its committee members had a defence finishing at the time this one is starting, up to the number of different members in a committee times the biggest value present in the list MW_m, which is the weight MWW, times the biggest value present in the list O_m, which is the weight OW, times the biggest value in the list CZW, if all of its members comply with the aforementioned conditions and are given the highest values in those parameters. Since these values are already included in the variable Z_{tdh} they do not need to be multiplied by in the objective as was seen in the previous two (4.17; 4.18).

$$Max \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \tag{4.19}$$

In order to be fair, the chairpersons of the committees, who have been prioritised in the past, the decision-maker has the ability to set a different weight MW_m for these members for the objectives (4.17), (4.18) and (4.19). Furthermore, there is also the possibility of setting different weights O_m for members who do not teach or often travel to the Taguspark *Campus* or for other reasons the decision-maker deems necessary.

4.3 Chapter Considerations

Within the last chapter, the problem was further described, and the implemented mathematical model was presented, including its sets, parameters, variables, constraints and objectives. Moreover, it is important to note that this is a multi-objective formulation, thus, it becomes necessary to implement a more complex solution approach than it would be in case only one objective was present. The chosen method will be addressed in the following chapter.

Furthermore, it should be noted that a number of characteristics of the model at hand were influenced by other works, previously introduced in the Literature Review. Besides the obvious parallels between the case at hand and other thesis defence scheduling problems, three other aspects were heavily inspired by the literature, namely the introduction of preference levels for time-slots for individual committee members, the compactness measure and some scheduling fairness concerns.

In the work of Fastré (2017), the authors also focused on the problem of thesis defence scheduling and a form of individual preference measure is also introduced, with the reserve day objective. Nonetheless, in the case at hand, the approach taken was more in line with works such as Vermuyten et al.

(2016) or Zhang et al. (2019), with different individual preference levels for each time-slot or other concerns being allowed within the model.

The implemented compactness measure, unlike the preference level measure, does not have an exactly similar approach in the literature, as it was specifically thought for the case at hand. Nonetheless, some inspiration was drawn from works from course timetabling problems (Beligiannis et al. 2008, Burke et al. 2010, Lü & Hao 2010, Lach & Luebbecke 2012, Bellio et al. 2016), which had the objective of reducing idle periods between consecutive courses.

Moreover, this formulation also takes into account fairness concerns, which have also been addressed previously. A distinction was made in Section 3.5 regarding works that either define fairness in scheduling problems as creating schedules that are as even as possible between each member, or works that define it as differentiating the affected based on their standing. In this formulation, influence from both groups can be seen. Firstly, the introduction of the possibility of setting different weights for individuals in different positions, which in the case at hand are MWW and OW, is the most common approach for works that intend to differentiate those individuals (Vermuyten et al. 2016, Fastré 2017). Furthermore, the incremental penalty seen in objective 4.18 is more often seen in the cases that see every individual as equal regarding scheduling weight (Penn et al. 2017, Battistutta et al. 2019), nonetheless, it is also found in a work belonging to the other group (Barnhart et al. 2012).

Chapter 5

Solution Approach

Within this chapter, the approach taken to solve the problem is explained. Firstly, the two-stage optimisation strategy is presented (Section 5.1) and, afterwards, the two approaches taken to solve the second stage are addressed, with Subsection 5.1.1 dealing with an *a priori* optimisation approach and Subsection 5.1.2 explaining the two stage augmented ϵ - constraint approach. Finally, a conclusion to the chapter, including parallels with the literature, is made in Section 5.2.

5.1 Two Stage Multi-objective Optimisation Strategy

For the case at hand, the objective of maximising the number of thesis defences scheduled (4.16) can be clearly identified as the primary one. Regardless of the values of the other three objectives (4.17)–(4.19), the first one must always have the highest possible value. To ensure this, a two stage optimisation approach was taken. During the first stage, the model finds the maximum number of thesis that can be scheduled and saves that value as a new parameter named TB. Then, in the second stage, a constraint setting that value for the number of thesis to be scheduled is added to the model, effectively turning that objective into a new hard constraint (5.1).

$$\sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} X_{tdh} = TB \tag{5.1}$$

Furthermore, two different strategies to solve the second stage of the model were implemented and made available to the decision-maker. The first takes an *a priori* approach to the multi-objective problem, whereas the second one uses a ϵ-constraint based approach, based on the works Mavrotas (2009) and Mavrotas & Florios (2013) and which the authors name AUGMECON. The main benefit of the first approach is that it is the most efficient one in terms of computing speed, allowing for larger instances to be solved in much faster times than the second approach. Nonetheless, the second approach presents the opportunity of gathering a multitude of solutions from the Pareto Front, leaving it up to the decision-maker, with full knowledge of that set of solutions, the choice of which is the preferred one.

5.1.1 *a Priori* Optimisation Approach - Second Stage

The first stage is, as mentioned before, to find how many thesis defences can be scheduled by maximising objective (4.16) and then set that value as a bound (5.1). After that part is complete, the model will then optimise the remaining three objectives (4.17)–(4.19).

As this is an approach based on an *a priori* method, it requires the decision-maker to indicate weights for each objective before the model is solved, and then, based on those combinations, calculate the optimal solution of a weighted single objective function. This allows the problem to be simplified, by turning its remaining three objectives into a single one (5.2) that can be maximised without any additional considerations.

$$\begin{aligned} Max \quad & HPW \times \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \\ & - GW \times \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} \\ & + ZW \times \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \end{aligned} \tag{5.2}$$

Thus, the following three weights were added to the model:

- HPW - Weight regarding the objective of maximising the preferences of each member regarding time slots (4.17);
- GW - Weight regarding the objective of minimising the number of days a member is scheduled on (4.18);
- ZW - Weight regarding the compactness of schedules objective (4.19)

Finally, a diagram summarising the complete two stage *a Priori* approach is presented in Figure 5.1.

5.1.2 Augmented ϵ - Constraint Approach - Second Stage

The second approach to solve the second stage is, as mentioned before, based on the AUGMECON approach introduced by Mavrotas (2009) and Mavrotas & Florios (2013), with some adjustments, that will allow a better fit to the case at hand. The benefit of using the augmented ϵ - constraint method is that it guarantees that every solution found is part of the Pareto set, which is not always true for conventional ϵ - constraint methods. Furthermore, it also introduces several other strategies to avoid redundant iterations.

We first illustrate the differences between the conventional ϵ-constraint method and the augmented ϵ-constraint method, using as an example a bi-objetive problem. Let $f1_x$ and $f2_x$ be the two objectives to be maximised. In the conventional ϵ - constraint method, to obtain solutions, the first step is to find the maximum values both objectives can take, which will be henceforth called, respectively, M1 and M2. From the optimal solution of each individual objective, we find a lower bound for the other objective, henceforth called N1 and N2. Then, following this method, new solutions are found by maximising either

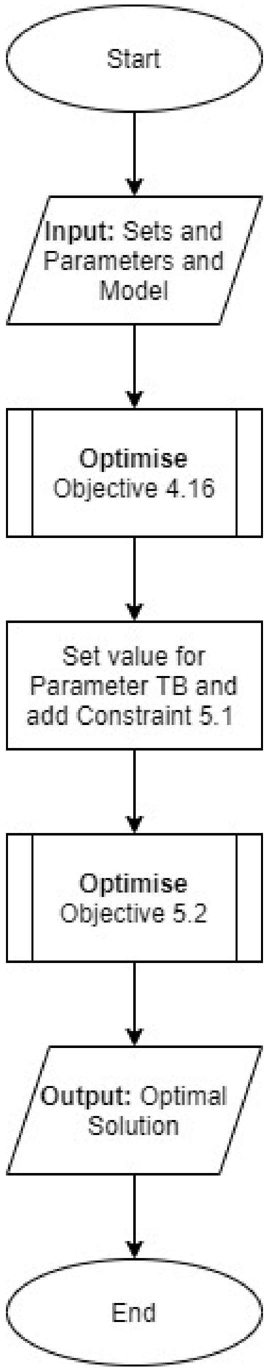


Figure 5.1: Two Stage *a Priori* Optimisation Approach Diagram

objective and setting a lower bound, LB_i, to the other one (5.3), equal to their initial lower bound, i.e., either N1 or N2 plus j times ϵ times the difference between the maximum value, either M1 or M2, and the initial lower bound, either N1 or N2, with j being the iteration number, and until LB_i is larger than M_i.

$$LB_i = \epsilon \times j \times (M_i - N_i) + N_i \tag{5.3}$$

Nonetheless, in some cases, this method can produce solutions that are not Pareto optimal. An example of this issue, addressed in Mavrotas (2009), is shown in Figure 5.2.

This happens because even if we set a lower bound for a certain objective $f2_x$, and maximise the

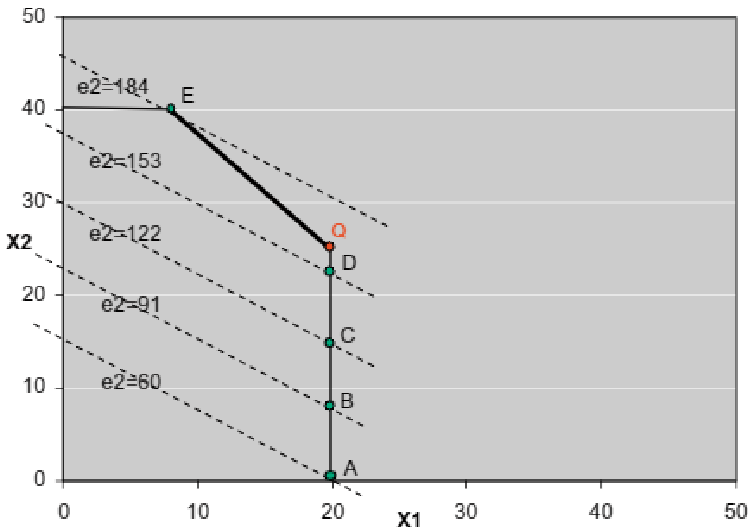


Figure 5.2: Results of the conventional ϵ - constraint method. Source: Mavrotas (2009).

best possible value for another objective $f1_x$ considering that lower bound, nothing guarantees that there is not another point, with a bigger value for $f2_x$, that has the same value for the other objective. In the example of Figure 5.2, this is true for points A, B, C and D, which have the best possible value for X1, but can be improved in objective X2, thus they are only weakly efficient solutions and not part of the Pareto optimal set, which is located between points Q and E.

The AUGMECON approach proposes a solution to this problem. Considering the previous notation, let s_i be a slack variable representing the difference between the best possible value for objective i and its value in iteration j. By including this variable in the objective function (5.4), that is minimising the slack of one objective while maximising the other, it is guaranteed that every solution the algorithm finds is part of the Pareto optimal set as for every optimal value of the first objective, the best possible value for the second objective is also found. Furthermore, another parameter eps, to be multiplied by the slacks, is added. This parameter is an adequately small number, usually between 10^{-6} and 10^{-3}, which guarantees that the second objective does not interfere with the value of the first one. Moreover, the method also eliminates redundant iterations by skipping the ones where the lower bounds for the objectives are smaller than values found in previously known solutions, as that iteration will result in a previously known solution. Beyond that, it is also possible to skip iterations when it is know that the set of lower bounds will produce an infeasible solution, even if this is only applicable in problems with more than two objectives.

$$f1_x - eps(s_2) \tag{5.4}$$

While the previous example was made with a bi-objective problem, the algorithm is applicable to instances with more objectives as well, such as the case at hand. Nonetheless, to implement the AUGMECON algorithm to the MEGI thesis defence problem, several new parameters and variables are

Table 5.1: New Parameters and Variables for the AUGMECON Implementation

Parameters	
$\epsilon \in \{0, 0.5\}$	
eps	Adequately small number
epss	Adequately small number
LBP	Lower bound for objective (4.17)
LBZ	Lower bound for objective (4.19)
MP	Best value for objective (4.17)
MZ	Best value for objective (4.19)
NP	Worst value for objective (4.17)
NZ	Worst value for objective (4.19)

Auxiliary Variables	
sp	Slack variable for objective (4.17)
sz	Slack variable for objective (4.19)

necessary.

As both objectives (4.17) and (4.19) are the ones being bounded through the AUGMECON method, the main objective to be maximised is the objective (4.18). The positions from each objective are interchangeable, nonetheless, since this was the only one with a negative value, for the sake of simplifying the problem this is how it was arbitrated.

First, the ϵ parameter must be defined. The algorithm was implemented considering a value greater than 0 and lower than or equal to 0.5, this is necessary as negative values for ϵ would not iterate properly and values greater than 0.5 would eliminate too many possible solutions and would lead to the iterations always stopping after the third one. Furthermore, while this is not necessary for the algorithm to run properly, the value of $\frac{1}{\epsilon}$ should be an integer number, so that the final iteration corresponds to a slack variable equal to 0. Moreover, as was previously stated, it will be necessary to know the maximum and set a minimum value for the objectives that will enter the objective function as a slack, as well as introducing the slack variables themselves. Lastly, new parameters regarding lower bounds for certain objectives will also be necessary. A summary of all the new parameters and variables can be found in Table 5.1.

Similarly to the previous *a priori* approach, the first stage is to find how many defences can be scheduled, and add constraint (5.1) to the model. Afterwards, the values for the new parameters MP, MZ, NP and NZ have to be found. To find the maximum values, two iterations, corresponding to the maximisation of objectives (4.17) and (4.19), are necessary. These values will be used to set the minimum values as well. Furthermore, it is necessary that these minimum values represent points in the Pareto Front.

To ensure that, the strategy that was taken when finding the worst values for objectives (4.17) or (4.19) was to set the value for the opposite objective as its maximum and then optimize the model considering as the objective function objective (4.18) plus either (4.17) or (4.19) multiplied by an adequately small number, epss, which, for the case at hand, was set as 10^{-4}. Thus, to find NP, constraint (5.6) is added to the model and objective (5.5) is maximised and, in turn, to find NZ, constraint (5.8) is added to the model and objective (5.7) is maximised.

$$Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} + epss \times \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \tag{5.5}$$

$$\sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} = MZ \tag{5.6}$$

$$Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} + epss \times \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \tag{5.7}$$

$$\sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} = MP \tag{5.8}$$

It is important to note that, through this strategy, NP and NZ are approximations of the Nadir points, which are defined as points that are still part of the Pareto Set while having the lowest possible value for certain objectives. In bi-objective problems, these points are usually easy to find. However, with more than two objectives, this becomes a more complex task. Still, in some cases, the adopted strategy might still find the Nadir points. This happens, for one of the objectives, when it is possible for the other two objectives to have their highest possible values within the same solution. Thus, when that is verified and the third objective cannot be improved, that is the Nadir point. Furthermore, it should also be noted that if that is not the case, the solutions between the Nadir Point and the N point are going to be lost, which, for the case at hand, is not a problem, as knowing the entire Pareto set is not the main goal and these solutions are most likely very poor in one of the objectives.

Before starting to search for solutions, it is also necessary to define the slack variables sp (constraint 5.9) and sz (constraint 5.10). They can be computed as the difference between the maximum values their respective objectives can take and the values they are taking in a given iteration.

$$sp = MP - \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \tag{5.9}$$

$$sz = MZ - \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \tag{5.10}$$

Finally, all the tools to define the objective function to be maximised in this method have been presented. The objective function 5.11 is the one being maximised in every iteration of this method. It can be divided in two parts, with the first one being the objective (4.18), which is not bounded and thus is the one being optimised, and the sum of the slack variables multiplied by eps, so that they do not interfere with the value of the other objective. Note that the slacks are divided by their respective maximum values, therefore, their sum will always be a number between 0 and 2, meaning that eps can be set as 10^{-1}, as that will be enough to guarantee that their values will not matter for the other part of the objective function.

$$Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} + eps \times (\frac{sz}{MZ} + \frac{sp}{MP}) \tag{5.11}$$

The method also has a specific mechanism to iterate between different solutions. Just like the conventional ϵ - constraint method, this mechanism is based on setting different combinations of lower bounds for the objectives, which, for the case at hand, are objectives (4.17) and (4.19), and their lower bounds are respectively LBP and LBZ. To guarantee that in each iteration they are respected, the following constraints are added to the model:

$$\sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \geq LBP \tag{5.12}$$

$$\sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \geq LBZ \tag{5.13}$$

Moreover, what guarantees that different solutions are searched is setting different values for LBP and LBZ in each iteration. This is achieved by adding ϵ times the difference between the maximum value of an objective and its worst value every time a new iteration starts. Once again, it is necessary to choose the order in which the lower bounds are incremented. At the start of the method, both lower bounds are equal to their respective worst values, then it was arbitrated that LBP is kept the same for the following iteration and LBZ is incremented by $\epsilon \times (MZ - NZ)$, in every iteration, until LBZ is greater than MZ, after which point its value is reset to NZ and LBP is incremented once by $\epsilon \times (MP - NP)$. Afterwards, LBZ is incremented again, until it is greater than MZ, and so on and so forth, until LBP is greater than MP and the algorithm stops. Note that changing the order between these iterations would not affect the overall efficiency of the algorithm.

It is possible to foresee the maximum number of solutions the algorithm can find based on the value of ϵ and the number of objectives, which in this case is three. If we represent the number of objectives as n, that number is $(\frac{1}{\epsilon} + 1)^{n-1}$, that is, for the case at hand, $(\frac{1}{\epsilon} + 1)^2$.

Furthermore, some of these iterations can be evaluated before the model optimisation phase starts, as it is possible, in some cases, to foresee if they will produce an equivalent result to one already obtained, this being true for both past solutions and combinations that were proven infeasible. In cases like those, the method implements strategies to skip those iterations. In particular, in cases where there has been a previous solution with both values for objectives (4.17) and (4.19) greater than the current corresponding lower bounds, LBP and LBZ, it is possible to foresee an equivalent result, thus, that iteration is skipped. Moreover, in instances where there has been an infeasible combination with LBP and LBZ smaller than or equal to the current ones, it is also possible to infer that the present iteration will produce another infeasible solution and can, therefore, be skipped.

A diagram of the Two Stage ϵ - Constraint Approach Diagram is presented in Figure 5.3.

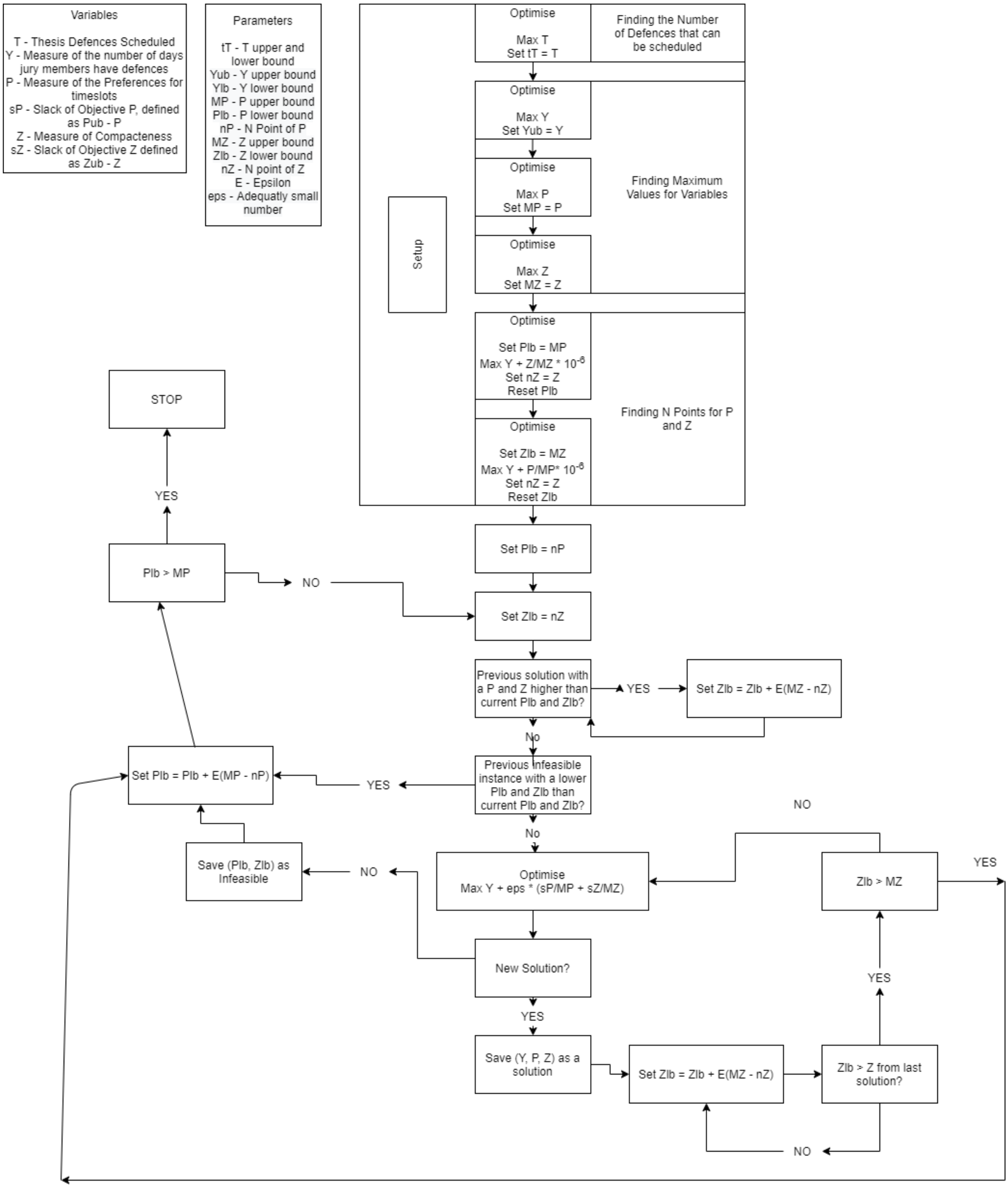


Figure 5.3: Two Stage ϵ - Constraint Approach Diagram

5.2 Chapter Considerations

Along this chapter, two different approaches to solve the problem were presented, the Two Stage *a Priori* Optimisation Approach and the Two Stage Augmented ϵ - Constraint Approach, with both having advantages and disadvantages.

The first stage is similar for both cases, with its goal being to find the maximum number of thesis defences that can be scheduled and setting that number as a hard constraint for the next stage. The main benefit of implementing this stage is that, for cases where some defences cannot be scheduled due to committee member availability or other concerns, the approaches can still present solutions, which would not be the case if it was simply stated that every defence must be scheduled. Furthermore, one could also consider the implementation of a single stage approach, with the objective 4.16 receiving a much higher weight than the other three. If this was the case in the *a Priori* approach, several problems could arise. On the one hand, if this weight was set too high, it could mean that the other objectives would simply be disregarded by the solver, whereas if it was too low, it could mean that the model would not present a solution with the maximum possible value of scheduled defences. On the other hand, it simply would not make sense to do that for the two stage augmented ϵ - constraint approach, as it would not only add a fourth objective and increase the complexity of the problem, but it would also mean that we would obtain solutions were not all possible thesis defences are scheduled, which is not the goal. This approach was inspired mainly by Vermuyten et al. (2016), which also employs a two-stage approach, but for which the main motivation was to make the model more efficient.

As for the second stage, the *a Priori* approach is the more common one in the literature, since it allows obtaining solutions quicker in comparison with *a Posteriori* methods, like the AUGMECON, as the latter have to generate several solutions and, thus, iterate between a set of optimisation phases. On the other hand, the AUGMECON, introduced by Mavrotas (2009) and Mavrotas & Florios (2013), provides several other advantages, such as allowing for a better knowledge of possible solutions before making a decision and having less subjective weights that the decision-maker must choose, as the weights HPW, GW and ZW are not necessary.

Finally, it is important to test and compare the results of both approaches in real-life or representative instances. Furthermore, in order to correctly make use of the tools made available, the decision-maker needs to take some concerns into consideration, which will be presented through a user guide in Chapter 7.

Chapter 6

Computational Experiments

In this chapter, the performance of each approach is evaluated. Firstly, the instance generator that was implemented is presented in Section 6.1 with the methods to generate the several necessary parameters being addressed in Subsections 6.1.1, 6.1.2 and 6.1.3, and the ensuing instances presented in Subsection 6.1.4. The results for the *a Priori* Two Stage approach is given in Section 6.2, with the first and second stage being addressed in Subsections 6.2.1 and 6.2.2, respectively. The effect of the available rooms and the compactness constraints is evaluated in Subsections 6.2.3 and 6.2.4. Lastly, the results for the Two Stage Augmented ϵ - Constraint Approach is analysed in Section 6.3. The analysis on its second stage initialisation is made in Subsection 6.3.1 and the number of effective iterations and the time it took to solve the instances is presented in Subsections 6.3.2 and 6.3.3, respectively.

All computational experiments were conducted using a Intel(R) Core(TM) i7-6500 CPU @ 2.50GHz 2.59GHz and 8GB of installed RAM. Moreover, the employed software was Python 3.7 and Gurobi 9.0.0.

6.1 Instance Generator

In this section, the method for creating instances for the problem will be explored. Several sets and parameters will serve as inputs to the instance generator, which will then create randomised data to serve as input to the model.

Three different parameters that need to be generated where identified. Firstly, there is the availability of the jury members, represented by the parameter HP_{mdh}, as well as the composition of each committee, represented by CM_{tp}. Lastly, there is the list MW_m, which represents different weights for committee members.

6.1.1 Jury Member Availability

The parameter HP_{mdh} represents the availability in each time slot (d, h) for member m. If the jury member is not available to start a defence at a certain time-slot, its value should be 0. On the contrary, if the member can be present, the time-slot should be assigned an integer number that represents the level of preference, with higher values representing a higher level of preference.

Furthermore, it is to be expected that the jury members' available or non-available time-slots occur in blocks. That is, if a member is available to start a defence for example at 9 am, it is very likely that he/she is also available to start one at 9:15 am, with the same being true for times when he/she is not available. Thus, it is possible to represent this relation as a conditional probability, where the probability of hour h being available or not, depends on the value for hour h - 1. Furthermore, the distribution of the values for the duration of an availability and unavailability block can be represented as a geometric distribution.

Consider that we represent the probability of having a value i for time-slot (d, h) given a value i' for the time-slot (d, h - 1) as $P(i(d,h) \mid i'(d,h-1))$. For the problem at hand 3 different values for i where considered, 0, 1 and 2. With that in mind, the probabilities for each value are as follows:

- $P(0(d,h) \mid 0(d,h-1)) = \alpha$
- $P(1(d,h) \mid 0(d,h-1)) = \frac{(1-\alpha)}{2}$
- $P(2(d,h) \mid 0(d,h-1)) = \frac{(1-\alpha)}{2}$
- $P(0(d,h) \mid 1(d,h-1)) = \alpha \times (1-\beta)$
- $P(1(d,h) \mid 1(d,h-1)) = \beta$
- $P(2(d,h) \mid 1(d,h-1)) = (1-\alpha) \times (1-\beta)$
- $P(0(d,h) \mid 2(d,h-1)) = \alpha \times (1-\beta)$
- $P(1(d,h) \mid 2(d,h-1)) = (1-\alpha) \times (1-\beta)$
- $P(2(d,h) \mid 2(d,h-1)) = \beta$

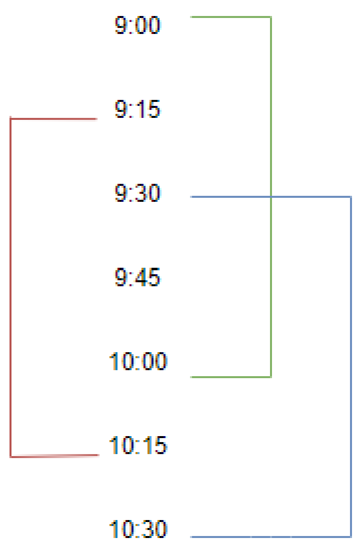


Figure 6.1: Example of Time-Slot Availability

As all the probabilities depend on the value for h - 1, it was considered that when h = 0 the value for h - 1 is 0. Moreover, it makes no sense, in the context of the problem, that a committee member is, for example, available to be present at a defence starting at 9 am, thus being available from 9 to 10 am (Fig 6.1, green line), since the defences are 1 hour long, and then being unavailable to start at 9:15 am (Fig 6.1, red line) and available again at 9:30 am (Fig 6.1, blue line). Therefore, when jury members have an available time-slot or block of time-slots and start being unavailable, they should be unavailable to start a defence for at least enough time-slots for a defence to be held. For example, if the jury member is available at 9 am, and not available at 9:15, he/she can only be available again at 10:15, thus having the time from 10 am to 10:15 am as the actual time when he/she is not available. Therefore, there needs to be an additional set of conditional probabilities, for when there already is an available time-slot in the day in question, where the probability of having a 0 on the three slots following the first 0 after an available sequence, is 100 per cent. On the other hand, there is also another set for when there has not been an available time-slot, which would be similar to the first one, but without the aforementioned four zeros rule. To test the results of this method, 10000 different days were randomly generated for different α and β combinations and the results are presented in Figure 6.2, where the amount of times a day has a certain time-slot marked as available is represented as a percentage of the total of randomly generated days, for each combination of α and β.

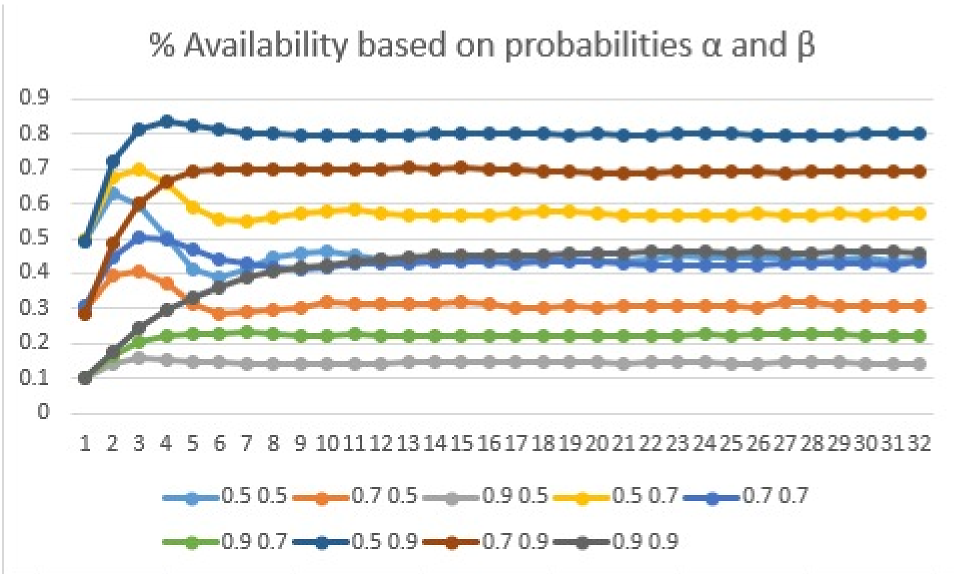


Figure 6.2: Availability percentage for each time-slot considering different α and β combinations

The distribution of availability created by this method is not uniform until after a certain point, where it stabilises. There are two different factors that explain this behaviour. Firstly, the fact that h - 1 for h = 0 was set to 0 means that the availability percentage for h = 0 is always 1 - α. Additionally, since the four zeros rule only comes online after there is a 1 or 2, the average duration of a block of unavailability if it starts at h = 0 can be computed as $1 + \frac{1}{1-\alpha}$, whereas its average duration becomes $4 + \frac{1}{1-\alpha}$ if it starts after an available time-slot. Thus, the average duration of unavailability blocks for the first few time-slots in a day should be smaller than after a certain point. To ensure that the results are easier to compare, for the instances generated in the future, the first 15 time-slots will be disregarded, so that the expected distribution of availability is uniform throughout the day.

6.1.2 Committee Composition

The committees are composed by three different members: the chairperson, the supervisor and an additional member. The group of professors that can be chairpersons, can also be supervisors and additional members, whereas the supervisors can also be considered for an additional member spot. Furthermore, the same member cannot be assigned two different positions within the same committee.

The method for generating committee compositions occurs in three steps, one for each position. Moreover, it needs as its input the number of thesis to be scheduled, the number of members np that can be assigned for all positions, the number of members ns that can be assigned as supervisors and additional members, and lastly the number na for those that can only be assigned as additional members. Often, the amount of thesis defences members have is balanced within the different categories, with some exceptions sporadically. Thus, to approximate this behaviour, the number of thesis defences assigned to members within each different position was represented as a normal distribution. Let also P(CCM(m,i)) be the probability of a member m being assigned to role i, with i = 0 representing the chairperson, i = 1 the supervisor and i = 2 the additional member, in a thesis defence.

To assign the chairpersons, first, a list of weights wp is created, with np different values, each corresponding to a member m $\in [0, np[$, which will have a truncated normal distribution, with 1 as its minimum value and with average $\sigma = 5$ and variance $\mu = 2$. Afterwards, each thesis defence is assigned a member m as its chairperson, with the probability for each member being assigned to each thesis being:

$$P(CCM(m,0)) = \frac{wp(m)}{\sum_{m1=0}^{np} wp(m1)}, m, m1 \in [0, np[\tag{6.1}$$

The process for assigning supervisors and additional members is similar to the previous one, with the difference that it is necessary to ensure the committees do not have repeated members. Thus, for assigning supervisors, firstly a weight ws is assigned to a member m $\in (0, np+ns)$ following a truncated normal distribution as before, with the difference being the list of weights is now ws. Then, for each thesis defence, a supervisor is assigned with probability P(CCM(m,1)):

$$P(CCM(m,1)) = \frac{ws(m)}{\sum_{m1=0}^{np+ns} ws(m1)}, m, m1 \in [0, np+ns[, m, m1 \neq CM_{t,0} \tag{6.2}$$

Lastly, the list of weights for additional members, created for member m $\in [0, np+ns+na[$ was called wa and the probability of a member being assigned to a thesis defence as an additional member is represented below.

$$P(CCM(m,2)) = \frac{wa(m)}{\sum_{m1=0}^{np+ns+na} wa(m1)}, m, m1 \in [0, np+ns+na[, m, m1 \neq CC(t,0), CM_{t,1} \tag{6.3}$$

Note that in the end, the number of effective members that are part of all defences does not necessarily have to be np + ns + na, as there is the possibility of some of them ending up not being picked.

6.1.3 Members with Additional Scheduling Weight

This is the last parameter that needs to be randomly generated. It represents the members m that are given an additional scheduling weight OW. Thus, each member goes through a Bernoulli trial with probability p of being given the additional weight OW and 1 - p of being assigned the value 1. This probability was arbitrated as being 10% for every instance.

6.1.4 Instances

To test the algorithms, several instances were created. All the instances were created for a time period of 20 days, as that is approximately the number of working days in a month, as well as 32 quarter-hours available each day for thesis defences to start. Three major groups of instances were created, the first one corresponding to instances with 30 defences, the second 40 defences and the last one with 50 defences. Moreover, within each of them, three other subgroups were created as well, with the first one having a percentage of available time for committee members of 20 percent, the second 45 percent and the last 70 percent. Furthermore, for each of these instance subgroups, three instances were randomly created with similar parameters in order to get a better assessment of how efficient the algorithms are at solving each group. These instance groups are summarised in Table 6.1, furthermore, 3 different instances were created for each group.

6.2 *a Priori* Two Stage Approach

The main goal of this section is to present and analyse the results of applying the *a Priori* Two Stage Approach to the generated instances and finding its limits regarding the maximum size after which it becomes too complex to be solved in a reasonable amount of time. For this analysis, to get a fair assessment, the following parameter values were considered for every instance:

- GW = 4
- HPW = 3
- ZW = 2
- R = 5
- CZ = (0)
- CZW = (1)
- OW = 2
- MWW = 2

Table 6.1: Summary of the Instance Groups

	Maximum Number of Members			Additional			**Availability Characteristics**			
Defences	Chairpersons	Supervisors	Additional Members	Weight %	σ	μ	α	β	Availability %	**Group**
	12	15	8	10	5	2	0.9	0.7	20	A1
30	12	15	8	10	5	2	0.9	0.9	45	A2
	12	15	8	10	5	2	0.7	0.9	70	A3
	12	20	13	10	5	2	0.9	0.7	20	B1
40	12	20	13	10	5	2	0.9	0.9	45	B2
	12	20	13	10	5	2	0.7	0.9	70	B3
	12	30	13	10	5	2	0.9	0.7	20	C1
50	12	30	13	10	5	2	0.9	0.9	45	C2
	12	30	13	10	5	2	0.7	0.9	70	C3

As previously stated, the value for the big-M is dependent on the values of weights OW, MWW, the highest value in the list CZW and the number of different positions, which is always three. Thus, its value can be computed as $BM = 2 \times 2 \times 1 \times 3 = 12$.

Additionally, a time limit of 3 hours was set for the resolution of each of the two stages. Moreover, for the first stage of the model, the number of thesis defences that could be scheduled will also be presented and, for the second stage, the remaining three objectives as well as the objective function value are presented.

A summary of the base instances results is presented in Table 6.2. Note that "G" represents the value for objective (4.18), "HP" for objective (4.17) and "Z" objective (4.19).

6.2.1 First Stage

As previously stated, the goal of the first stage is to find how many thesis defences can be scheduled for a certain instance so that this value can be set as a hard constraint for the second stage.

Out of the 27 instances tested, in 23 it was possible to schedule the entirety of the discussions. Furthermore, in every instance where this was not possible only one of the thesis defences remained unscheduled, which happened in instance 1 and 3 for group A1 and instance 1 and 3 for group C1. Both of those groups have the smallest availability percentage. These results prove the usefulness of regarding the number of defences scheduled as an objective instead of a hard constraint upfront, as, otherwise, these instances would simply be infeasible.

Every first stage optimisation took less than 1 minute, and, consequently, it was always possible to reach a gap of 0% within the time limit. Nonetheless, it is possible to note that both the increase in number of defences to schedule, and the availability percentage had an effect on the time that it took to solve the instances.

Table 6.2: Summary of the Base Instances Results

		1st Stage		2nd Stage					
Group	**Instance**	Scheduled Defences	Time (s)	G	HP	Z	Obj Function	Time (s)	Gap
A1	1	29	10	-392	234	8	-850	10	0
	2	30	2	-329	238	11	-580	2	0
	3	29	2	-258	236	13	-298	1	0
	avg	29.3	4.7	-319	236	10.7	-576	4.3	0
A2	1	30	4	-144	260	54	312	165	0
	2	30	3	-111	231	47	343	216	0
	3	30	3	-121	230	45	296	25	0
	avg	30	3.3	-125.3	240.3	48.7	320	135.3	0
A3	1	30	23	-60	262	56	658	10499	0
	2	30	23	-58	261	75	701	10800	4.7
	3	30	13	-46	251	68	705	2843	0
	avg	30	19.7	-54.7	258	66.3	688	8047.3	1.6
B1	1	40	25	-499	294	12	-1090	4	0
	2	40	13	-491	311	4	-1023	5	0
	3	40	10	-442	292	13	-866	8	0
	avg	40	16	-477.3	299	9.7	-993	5.7	0
B2	1	40	22	-169	327	50	405	851	0
	2	40	6	-154	344	64	544	114	0
	3	40	14	-149	320	67	498	50	0
	avg	40	14	-157.3	330.3	60.3	482.3	338.3	0
B3	1	40	49	-78	304	80	760	10800	14.2
	2	40	29	-96	328	80	760	10800	16.7
	3	40	48	-96	347	92	841	10800	16.7
	avg	40	42	-90	326.3	84	787	10800	15.9
C1	1	49	19	-638	393	16	-1341	5	0
	2	50	22	-608	412	28	-1140	8	0
	3	49	19	-559	363	21	-1105	8	0
	avg	49.3	20	-601.7	389.3	21.7	-1195.3	7	0
C2	1	50	22	-205	395	75	515	526	0
	2	50	33	-211	442	77	636	1066	0
	3	50	36	-189	368	80	508	8225	0
	avg	50	30.3	-201.7	401.7	77.3	553	3272.3	0
C3	1	50	55	-113	370	90	838	10800	23.5
	2	50	39	-109	387	99	923	10800	14.9
	3	50	53	-138	484	123	1146	10800	16.6
	avg	50	49	-120	413.7	104	969	10800	18.3

6.2.2 Second Stage

It is during the second stage that the algorithm optimises the weighted objective function with the remaining three objectives.

In 20 out of the 27 instances, it was able to achieve that within the 3 hours time limit. While every instance with 45 availability percentage or less was solved, on the other hand, none of the instances in group B3 and C3 were able to reach a gap of 0%, whereas for group A3, instance 2 was also not solvable within the time limit. The results point to the expected conclusion that the number of thesis to be scheduled influences the time it takes to solve the model, with larger numbers making it harder to computationally solve it, moreover, instances with more availability percentage took less time to solve, as it is easier for the model to find solutions. Furthermore, the time it took to solve the instances had a high variance even within the same groups, meaning that even for instances that look similar at first sight, there might be some differences that make it affect their solve times.

Within each major group, that is A, B and C, we can see that for the time-slot preference objective (4.17) there was not a noticeable increase in performance with the increase of the availability percentage, especially between instances with 45 and 70 percent. On the contrary, the other two objectives saw considerable improvements with the increase of the aforementioned parameter.

6.2.3 Number of Rooms

While for the previous experiments the number of rooms was set to 5, this is a parameter that affects the overall efficiency of the approach. For this reason, within this subsection, this effect will be assessed. The results for Group A2 are presented in Table 6.3 and for Group B2 in Table 6.4.

Table 6.3: Effect of Room Quantity on Instances from Group A2

		1st Stage		**2nd Stage**					
Group	**Instance**	Scheduled Defences	Time (s)	G	HP	Z	Obj Function	Time (s)	Gap
				1 Room					
	1	30	21	-193	292	42	188	1756	0
A2	2	30	6	-139	244	40	256	273	0
	3	30	4	-158	248	27	166	491	0
	avg	30	10.3	-163.3	261.3	36.3	203.3	840	0
				3 Rooms					
	1	30	4	-144	260	54	312	236	0
A2	2	30	6	-111	231	47	343	308	0
	3	30	4	-121	230	45	296	15	0
	avg	30	4.7	-125.3	240.3	48.7	320	186.3	0
				Base Instances - 5 Rooms					
	1	30	4	-144	260	54	312	165	0
A2	2	30	3	-111	231	47	343	216	0
	3	30	3	-121	230	45	296	25	0
	avg	30	3.3	-125.3	240.3	48.7	320	135.3	0
				7 Rooms					
	1	30	3	-144	260	54	312	252	0
A2	2	30	11	-111	231	47	343	165	0
	3	30	4	-121	230	45	296	18	0
	avg	30	6	-125.3	240.3	48.7	320	145	

Table 6.4: Effect of Room Quantity on Instances from Group B2

Group	Instance	1st Stage		2nd Stage					
		Scheduled Defences	Time (s)	G	HP	Z	Obj Function	Time (s)	Gap
1 Room									
B2	1	40	32	-213	325	50	223	10800	35.8
	2	40	8	-208	336	56	288	10800	21.9
	3	40	6	-208	332	57	278	6590	0
	avg	40	15.3	-209.7	331	54.3	263	9396.7	19.2
3 Rooms									
B2	1	40	24	-172	325	56	399	856	0
	2	40	7	-154	345	62	543	167	0
	3	40	6	-152	322	58	474	329	0
	avg	40	12.3	-159.3	330.7	58.7	472	450.7	0
Base Instances - 5 Rooms									
B2	1	40	22	-169	327	50	405	851	0
	2	40	6	-154	344	64	544	114	0
	3	40	14	-149	320	67	498	50	0
	avg	40	14	-157.3	330.3	60.3	482.3	338.3	0
7 Rooms									
B2	1	40	19	-169	327	50	405	928	0
	2	40	16	-154	344	64	544	127	0
	3	40	38	-149	320	67	498	45	0
	avg	40	24.3	-157.3	330.3	60.3	482.3	366.7	0

Firstly, we can see that for the tested instances, reducing the number of available rooms never meant that more defences would go unscheduled, instead, it simply led to more cases where the committee members would have defences scheduled on more days, which led to a decrease in the value of objective G. Nonetheless, it is possible that, in other cases, with less available days or less availability from the members, that this decrease would lead to a decrease in the number of defences scheduled.

Out of all the tested scenarios, two, both in Group B2 and with one room, were not solvable within three hours. Furthermore, this group had the highest average gap out of every other scenario. Moreover, it is possible to note a tendency for a reduced model efficiency with the reduction of the number of rooms, especially when that leads to decreases in the value of the objective function and overall quality of the solution, this points to the conclusion that since there are fewer rooms, it becomes harder to find suitable schedules and therefore the computer took a longer time to solve the problems.

As expected, reducing the number of rooms leads to worse values for the objectives in general. Still, there were some scenarios where the decrease in number of rooms lead to a better result in objective HP. This happens because there is a trade-off between fulfilling the preferences for time-slots and minimising the number of days members have defences scheduled on and their compactness and, while the decrease in the number of rooms always led to an equal or worse performance in those two objectives, sometimes it is possible to then try to have a better performance in the other one. Nonetheless, for all tested instances, there was no difference in the results for 5 and 7 rooms in terms of objective value and marginal differences in terms of efficiency. This happens because there was never a scenario where more than 5 defences being scheduled at the same time would lead to a better solution with the given parameters.

6.2.4 Compactness Constraints

One of the aspects that has the greatest effect in the time it takes to solve the instances is the value of the big-M related to the compactness objective. In this subsection, this effect will be assessed. Different values for big-M depend on the value of parameters OW and MWW, the highest value of CZW and the number of different committee member positions. The possibilities that will be analysed are displayed in Table 6.5.

Table 6.5: Different Big-M Values

Big-M	**OW**	**MWW**	**CZW**	**Positions**
0	-	-	-	-
3	1	1	{1}	3
6	1	2	{1}	3
12	2	2	{1}	3
24	2	2	{2, 1}	3
36	2	2	{3, 2, 1}	3

The first value for the big-M corresponds to removing the compactness constraints and objective from the formulation. The example with the value 3 translates into the same model but without distinction between either chairpersons and the additional weight and the other members, whereas the example with value 6 already incorporates the weight for the chairpersons and the value 12, the one used in the base instances, has both weights included. The values 24 and 36 represent the enlargement of the compactness zones and the setting of their correspondent weights, to a maximum of 2 or 3, respectively. The results of varying the values for the big-M parameter are shown in Table 6.6. Within the aforementioned table, the time it takes to solve certain instances or based on the value for the big - M is presented, furthermore, in cases where it was not possible to solve the instance within the 3 hours time limit the reached gap is presented instead.

For most cases, the increase in this value leads to a higher time to solve the instance. Furthermore, there were two cases, both in Group B2, where the increase to the maximum tested value led to the instances not being solvable within the three hour time limit.

Moreover, an increase from 12 to 24 and from 24 to 36, always led to an increase in the time it took to solve the instances, meaning that the decision-maker must decide on the trade-off between having more differentiation on the weights employed. This is one of the limitations of the big - M method, as it can in some cases lead to inefficiencies on the branch and bound solvers. On the contrary, the same is not true for the increase from 3 to 6 and 6 to 12. This means that the values from the weights are important for the model to distinguish between solutions that would otherwise be equivalent and, sometimes, this effect ends up being more beneficial to the efficiency of the formulation than the reduction in the value for the big-M.

Table 6.6: Effect of Different Big-M Values on the Time to Solve the Second Stage for Instances of Groups A2 and B2(s)

		Big-M					
Group	**Instance**	0	3	6	12	24	36
A2	1	26	196	160	165	590	1794
	2	30	326	201	216	1795	4256
	3	8	14	11	25	30	105
B2	1	31	557	675	851	6700	15%
	2	19	330	397	114	731	3.8%
	3	14	55	165	50	640	2055

6.3 Two Stage Augmented ϵ - Constraint Approach

The main goal of this section is to understand how efficient the Two Stage Augmented ϵ - Constraint Approach is at solving different data sets and finding its limits regarding the maximum size after which it becomes too complex to be solved in a reasonable amount of time. The first stage of both approaches is the same, thus, there is no need to test it again. Furthermore, each iteration of the Augmented ϵ - Constraint Approach is similar to the second stage of the *a Priori* Approach, with the added lower bounds to the objectives. For this reason and since it is possible to know the maximum number of iterations, it is possible to give a rough estimation of the maximum time it would take to solve each instance, which is useful to test the time gains originating from the implemented skipping mechanisms. Moreover, note that for the larger instances, that took longer to solve, this second approach is not appropriate, as that time will multiplicably increase by the number of iterations, and it will not be possible to solve them within a reasonable time-frame. Thus, only instances from groups A1 and A2 were tested.

6.3.1 Second Stage Initialisation

The first part of the second stage of the approach is to find a starting point to set the lower bounds for the two objectives that are being bounded. In some cases those values might be the Nadir points, thus, an analysis of the efficacy of the approach in finding those points will be made.

A summary of the results can be found in Table 6.7. This table is divided into three sections that reference the three substages of the initialisation phase. The first section "Maximum Values" shows the time it took to find the maximum possible values for each objective and that value. Then, in the following two sections, it presents the time it takes to find both N points and the corresponding values for each of the objectives.

It was only possible to be sure that a lower bound point was the Nadir point for that objective once. That only happened because the value for the variable was 0, thus, it would not be possible for it to be

Table 6.7: Initial Values for Lower and Upper Bounds

		Maximum Values						**N Point HP**				**N point Z**			
Group	**Instance**	G	T(s)	HP	T(s)	Z	T(s)	G	**HP**	Z	T(s)	G	HP	**Z**	T(s)
A1	1	-386	2	259	3	13	3	-452	225	13	3	-560	259	0	5
	2	-329	2	268	2	12	2	-342	242	12	2	-543	268	6	4
	3	-257	4	264	2	15	2	-290	239	15	2	428	264	2	6
A2	1	-137	20	339	4	75	701	-168	244	75	365	-452	339	21	11
	2	-107	70	285	3	68	321	-141	205	68	1591	-297	285	20	4
	3	-121	10	285	3	61	200	-177	220	61	480	-308	285	24	4

even lower, whereas for the other ones, since the value of the remaining two objectives were not their maximum values it could not be proven that those points were the Nadir points. Still, the main goal of this method was to, in reasonable time, find an acceptable lower bound to start from and not to find the Nadir points with certainty.

In instances from group A1 there was not a very high variance between the time it took to find each point, furthermore, every one of them only took a few seconds to find. On the contrary, in group A2, there was not only significant variance in the time it took to find the different points for the same instance, but also between the time it took to find equivalent points for different instances. Increasing the committee member availability percentage, in general, has that same effect in both the maximum and n point values. Additionally, it also increases the time it takes to find the aforementioned points.

While all the points for group A1 were found within 6 seconds, the same is not true for larger instances, that is, the group A2. Two specific points seem considerably harder to find, the maximum value for Z and the N point for HP. What they have in common is that in both, the main objective being maximised is Z, which is the one associated with a big-M formulation, thus, we can conclude that this objective may be the main bottleneck when solving larger instances.

6.3.2 Skipped Iterations and Number of Solutions

The approach skips iterations when either of two conditions are met, as was mentioned before. Firstly, if there is an iteration where it is possible to foresee an already found solution and secondly, if it is possible to know beforehand that an iteration will result in an infeasible solution. Thus, it is important to analyse, given a certain value for ϵ, how many solutions are found and how many iterations are skipped, compared to the maximum number of possible solutions that can be found.

The results are summarised in Table 6.8.

Table 6.8: Solutions and Skipped Iterations

		Effective Iterations		**Skipped Iterations**	
Group	**Instance**	Solutions	Infeasible	Feasible	Infeasible
		$\epsilon = \frac{1}{6}$ Iterations = 49			
A1	1	22	4	4	19
	2	10	1	0	38
	3	17	4	0	28
	avg	16.3	3	1.3	28.3
A2	1	22	4	0	23
	2	22	4	4	19
	3	21	5	0	23
	avg	21.7	4.3	1.3	21.7
		$\epsilon = \frac{1}{4}$ Iterations = 25			
A1	1	11	3	3	8
	2	7	1	0	17
	3	12	4	0	9
	avg	10	2.7	1	11.3
A2	1	12	3	0	10
	2	13	3	3	6
	3	13	3	0	9
	avg	12.7	3	1	8.3
		$\epsilon = \frac{1}{2}$ Iterations = 9			
A1	1	5	2	0	2
	2	4	1	0	4
	3	5	2	0	2
	avg	4	1.7	0	2.7
A2	1	5	2	0	2
	2	6	2	0	1
	3	5	2	0	2
	avg	5	2	0	2

In the tested instances, the reduction of the availability percentage, in general, led to a slightly smaller number of effective iterations, proving that reducing the available times-slots for committee members leads to less feasible schedules and, in consequence, less effective iterations. Furthermore, both the average number of effective solutions and infeasible iterations decreases. On the contrary, the number of skipped iterations increases with the decrease in availability. Moreover, the infeasible skipped iterations had a much higher occurrence than the feasible skipped iterations, meaning that the mechanism to skip infeasible solutions ends up gaining considerably more time than the other one.

6.3.3 Time to Solve each Iteration

The objectives will be given increasing lower bounds during each iteration of the second stage. Thus, even though the model is almost the same, the iterations might have very different times to solve. Therefore, the goal of this section is to understand the overall efficiency of the AUGMECON algorithms. A summary of these results is presented in Table 6.9, with the total and average times for each effective iteration, being presented, as well as the estimated time gained, which was calculated by multiplying the average time of effective iterations by the number of skipped ones, represented in Table 6.8.

While reducing the number of possible iterations always led to a decrease in the total time it took to solve the instance, there was not a noticeable trend in the average iteration time. Additionally, following a similar trend to the *a-Priori* Approach, we can verify that the instance group with the highest availability percentage, in this case A2, took the longest to solve, with one instance, namely the second instance in group A2 taking up to three hours to solve, including the initialization time, whereas the maximum time an instance of this group had taken with the first approach was of only three and a half minutes.

Comparing the results from both approaches, for the group A1, which was less complex to solve, the average time to solve the iterations was similar to the time it took to solve the same instances with the first approach. However, for group A2 there was a considerable increase in the average iteration time when compared to the first results. Furthermore, while the iteration time for group A1 had little to no variance, the same cannot be said for group A2. That behaviour is represented in Figure 6.3. In this figure, each colour represents one of the instances of Group A2. Then, the x-axis represents each effective iteration and the y-axis the time it took to solve it.

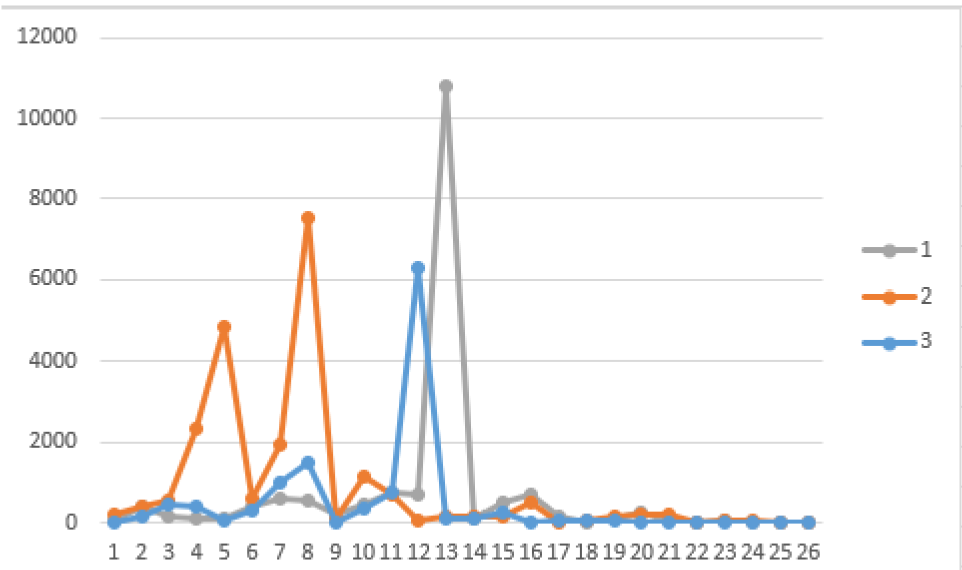


Figure 6.3: Time to Solve Each Effective Iteration for Each Instance for Group A2 with ϵ = 1/6

A defined trend can be found in the behaviour of the time it takes to solve each iteration. As previously explained, the proposed augmented ϵ - constraint approach increases the lower bounds for two of the three objectives, namely the compactness and time-slot preference objectives. Firstly, both lower bounds start at their n points and then the compactness objective is increased until it reaches

Table 6.9: Average Time per Effective Iteration and Estimated Time Gained Through Skipping Iterations

		Effective Iteration Time					
		Solutions		Infeasible		Estimated Time Gained	
Group	Instance	Average	Total	Average	Total	Feasible	Infeasible
$\epsilon = \frac{1}{6}$ Iterations = 49							
A1	1	4.4	97	4.3	17	17.6	81.7
	2	4	40	4	4	0	152
	3	4.6	79	5	20	0	140
	avg	4.3	72	4.4	13.7	5.9	124.5
A2	1	288	6335	2718.5	10874	0	62525
	2	636.3	13999	2016	8064	2545.2	38304
	3	185	3886	1614	8070	0	37122
	avg	369.7	8073.3	2116.2	9002.7	848.4	45983.7
$\epsilon = \frac{1}{4}$ Iterations = 25							
A1	1	4	44	4	12	12	32
	2	4	28	4	4	0	68
	3	4	48	4	16	0	36
	avg	4	40	4	10.7	4	45.3
A2	1	452.6	5431	266.3	799	0	7990
	2	1038	13494	2200	6600	3114	13200
	3	334.4	4347	98.7	296	0	888
	avg	608.3	7757.3	855	2565	1038	7359.3
$\epsilon = \frac{1}{2}$ Iterations = 9							
A1	1	4.2	21	4	8	0	8
	2	4.75	19	4	4	0	19
	3	2.8	14	4	8	0	8
	avg	3.9	18	4	6.7	0	11.7
A2	1	169	845	287	574	0	574
	2	1398	8390	40	80	0	40
	3	119.2	596	7	14	0	14
	avg	562	3277	111.3	222.7	0	209.3

its maximum value or there is an infeasible iteration, after which the value for the lower bound for the compactness objective is reset to n and the time-slot objective lower bound is increased, and the compactness objective starts being incremented again and so on and so forth until both maximum values are reached. Up to a certain point, the increase in the time to solve each effective iteration is directly linked to the aforementioned lower bound increments, with the time to solve increasing up to the point where the compactness objective is reset, after which the time to solve drops again. That behaviour then stops after the first infeasible iteration is reached, after which the high variance stops and the times with each compactness objective lower bound reset start decreasing instead of increasing as was seen up until this point. In the first part of this behaviour, what can be concluded is that the increased difficulty in finding solutions coming from the increasing lower bounds leads to larger computational solve times, whereas in the second part, where the computational times stabilise and are almost instant, the fact that there are fewer possible solutions has the opposite effect, and makes the model faster.

Lastly, it is possible to assess the usefulness of the iteration skipping mechanisms that were implemented. To do that, a rough estimation of the time gained through each of them is also presented in Table 6.9. To obtain these values, the average time it took for every feasible and infeasible iteration was multiplied by the number of feasible and infeasible iterations that were skipped, numbers which were presented in Table 6.8. While not many feasible solutions were ever skipped, and thus that mechanism did not produce efficiency gains in most instances, the infeasible skipped iterations could have meant an additional solving time of up to 17 hours for Instance A2 with $\epsilon = \frac{1}{6}$.

6.4 Chapter Considerations

The goal of this chapter was to evaluate the effectiveness of both solution approaches presented in Chapter 5, namely the *a Priori* Two Stage Approach and the Two Stage Augmented ϵ - Constraint Approach, at solving several types of instances. To do that, the first step was to implement an instance generator that would generate instances that would follow similar behaviours to the expected from real world ones.

The first stage of each approach is the same and, thus, it was only analysed in one of the sections. Moreover, this stage was never an issue in computational terms, as it was always solvable within less than a minute for every instance tested. Furthermore, its usefulness was proven as for some instances not all the defences were possible to be scheduled.

The second stage of the *a Priori* Two Stage Approach was able to solve most instances within the given time frame. Nonetheless, the increase in committee member availability proved to be challenging for the model. Moreover, decreases in available rooms also usually led to smaller solution times. Lastly, it was proven that the big-M employed in the compactness objective had a considerable effect in the time it took for the model to solve the instances, with certain increases turning some instances that were solvable within a reasonable time-frame into instances that would need more than the three hours time limit to be solved.

While providing different results, the second approach follows similar patterns and rules to the first one, thus, it was not necessary to test every instance that was tested for the first approach, as it would be possible to scale most results to the second one.

While solving the initialisation stage in this approach, it was possible to confirm that the compactness objective was definitely the main detriment to the algorithm's efficiency, as finding its maximum value as well as the HP N point, where effectively the compactness objective is also being maximised, were the hardest points to find regarding computational times.

As this approach iterates through the model several times, it takes considerably longer times to solve than the first one. Additionally, while for simpler instances this is not very noticeable, for larger instances the average iteration times were also larger than the time it took to solve the instance in the first approach. Meaning that this method is best suited only for simpler instances. On the other hand, the usefulness of the skipping mechanisms was proven, as, for some instances, up to 17 hours of infeasible iterations were roughly estimated to be skipped.

To obtain these results, the models were implemented using the solver Gurobi and Python. The guide to use the tools that were created is presented in Appendix A.

Chapter 7

Conclusions and Future Work

In this chapter the general conclusions from this work is presented and, afterwards, the possible areas where it might need improving on in the future to better fit the necessities of the decision-maker are discussed.

7.1 Conclusion

The scheduling of thesis defences in the Department of Engineering and Management at Instituto Superior Técnico is the responsibility of the department's secretary, who must coordinate with all the committee members and find the most suitable dates and times to have each defence on. This can prove to be an arduous process, depending on the number of defences to be scheduled at a particular time. Furthermore, due to the deadlines for submitting the dissertations, this person usually has two major workload peaks per year, during the ends of the first and second semesters.

While there is already some literature on this topic, it is a relatively recent subject of interest within the Academic Scheduling field, whereas subjects such as course timetabling and exam scheduling have a far larger representation. Still, there are some aspects found in previous works in the aforementioned field that were applied in this dissertation. To begin with, objectives such as ensuring the compactness of schedules or minimising the number of defences that would go unscheduled were already present in the thesis defence scheduling literature, whereas the individual preference for time-slots is only found in works from the other subjects in the Academic Scheduling field. Furthermore, the consideration of fairness in scheduling is also present in the aforementioned literature, with two main groups of scheduling concerns being found. The most prevalent is the one that considers as the fairest schedule the one that is the most balanced out and even between its members, and the other one being the one that considers differences in standing between them and tries to differentiate the different member's scheduling priorities.

Based on the characteristics of the problem at hand and taking some inspiration on several other works in the literature, a mixed integer linear programming model to represent our case was formulated, with reference to its sets, parameters, decision variables and constraints.

Four different objectives were considered to test the quality of the found solutions. The first one is the minimisation of the thesis defences that would go unscheduled. This objective is often modelled in the literature as a hard constraint, however a choice was made to represent it as an objective instead, as it would allow an incomplete schedule to be presented even if some committee members do not have aligning availabilities. Thus, instead of not presenting a solution, it is possible to present one without the defences that were not possible to schedule. The second objective was inspired by works on other Academic Scheduling problems. It aims to give the committee members the liberty of giving different preference levels to their available time-slots, with the model then trying to assign them to their preferred ones. The chosen notation was to assign different integer values as levels of preference, with a value of 0 being an unavailable slot and then the higher the value the higher the preference for a certain time-slot. The third objective comprises the minimisation of the number of days committee members have defences scheduled on. This is a concern as most committee members do not usually travel to the Taguspark *Campus*, thus they would rather have their defences in as few days as possible, so as not to have to make additional travels. Lastly, the final objective was to ensure that within the same day, the schedule for a certain committee member is as compact as possible. While there was some concern with this question in the thesis scheduling literature, it ends up being translated more into objectives more similar to our third one than to the one proposed in this dissertation. A compactness indicator was created to represent this concern. It measures how many of the committee members had a defence end within a certain time-frame before the start of another defence, then this indicator is maximised, so that the defences are as chained as much as possible for each committee member.

Furthermore, there was also a concern in ensuring that the schedule was fair to every committee member and that some of them were not overly harmed for the benefit of the others, moreover, the solution should not lead to lowering the quality of some schedules in the name of creating more balanced ones. Inspiration was taken from the two fairness groups referenced above. On the one hand, different weights were created to differentiate between committee members with different positions, namely the chairpersons, who usually have the most thesis defences to attend to. Additionally, the decision-maker can also opt for assigning a second weight for other members whenever this is deemed necessary. Note that both weights influence all the aforementioned objectives. On the other hand, there was a concern in ensuring that there was not a very high discrepancy between the number of days different members had to go to the Taguspark *Campus*, thus, instead of just minimising this number of days, an exponential penalty is included for every additional day a member is scheduled to attend a defence on, making it less likely that a specific member ends up with an excessive number of days scheduled for.

As the mathematical model had multiple objectives, it was necessary to find methods to solve it that would account for this, with two distinct approaches being proposed. Both approaches were divided into two stages, with the first stage being similar. In this stage the objective of scheduling as many defences as possible is maximised, this ensures that we can then set the maximum number of thesis that can be scheduled as a hard constraint in the following stage, guaranteeing that every thesis defence that can be scheduled is in fact scheduled. Then the second stage is what differentiates both methods, as they tackle the multi-objective nature of the problem differently.

The first approach was named Two Stage *a Priori* Approach. In it, the decision-maker must define weights for each of the three remaining objectives, which are then aggregated in a weighted objective function, with the final solution being the optimal solution for those three weights. The second proposed method was named Two Stage Augmented ϵ - Constraint Approach which is based on the ϵ - constraint method and, as such, instead of trying to evaluate and present the optimal solution, it is a strategy to search the solution space for feasible solutions. The adopted approach has the advantage, when compared to the traditional ϵ - constraint method, that it searches only for Pareto optimal solutions and has several mechanisms that promote the skipping of unnecessary iterations, as it evaluates *a priori* if they will result in an already known solution by comparing the set of lower bounds to the objective values of previous solutions or in an infeasible one by checking if there was a previous infeasible solution with smaller lower bounds than the current ones. Both approaches have advantages and disadvantages when compared to each other. The Two Stage *a Priori* Approach is considerably faster in its resolution than the Augmented ϵ - Constraint Approach, as it will only solve the model twice, once for each stage, whereas the second will have multiple iterations, depending on the value for the ϵ. On the other hand, the Augmented ϵ - Constraint Approach provides a more complete notion of the solution space and suggests several possible solutions for the decision-maker to choose from without the necessity of inputting weights for each objective.

To test both approaches, an instance generator was created and several instances with varying dimensions were randomly generated. The computational experiments allowed to draw several conclusions related to the functioning of both solution approaches. For one, the usefulness of the first stage was proven, as it allowed the scheduling of several instances where one of the defences could not be scheduled due to incompatible availability between committee members. Furthermore, this stage never took more than a minute to solve, proving its efficiency. The effect of varying several parameters was also tested, with some variations leading to considerably longer solve times, it is then left to the decision-makers which parameters are the best for their preferences.

A tool to facilitate the input and output of data to the model was also created. This tool uses Excel for both of those ends, with the necessary data being imported by the Python file and then the resulting schedules being exported to a different Excel sheet.

7.2 Limitations and Future Work

Lastly, in this section the limitations of this work and future work suggestions to improve it will be presented. Furthermore, it should be stated that this dissertation was done during the Covid-19 Pandemic, which brought several hindrances which will be addressed later.

For starters, it was not possible to apply either approach to real world instances, with all the computational experiments being made with randomly generated instances and, while these instances were engineered to follow similar rules and patterns to those that would be found in the real word, there might always be some exceptions and problems that did not occur in the generated instances and might happen in real instances. Following the aforementioned limitation, the proposed tool to facilitate the importation of data to the model as well as the presentation of the final schedules was also not tested by a third party, meaning that there was no feedback regarding its user-friendliness.

Thus, the main goal going forward would be to test both approaches using a real world instance and organise, together with the decision-maker, the scheduling of thesis defences during one of the peaks in defences to be scheduled in the end of the semesters. Then, it would be possible to monitor both the objective improvement of the quality of the schedules regarding characteristics such as the number of days committee members have defences scheduled on as well as the subjective level of satisfaction of the committee members with the new scheduling process and compare their perceived quality of the schedules proposed by the MILP models versus the quality of the schedules created using the current approach. Furthermore, with their feedback, it would be possible to improve both the scheduling process and add features to the model that they deem important, for example a process that would automate the input of data to the model such as the compositions of the committees or the availability of their respective members.

Furthermore, it would be interesting to study a method to help the decision-maker to choose between the different presented Pareto optimal solutions given by the Two Stage Augmented ϵ - Constraint Approach, as, as it currently stands, it only presents the decision-maker with a set of possible solutions, which, in some cases might only vary slightly.

Bibliography

Achá, R. A. & Nieuwenhuis, R. (2014), 'Curriculum-based course timetabling with SAT and MaxSAT', *Annals of Operations Research* **218**(1), 71–91.

Akkan, C. & Gulcu, A. (2018), 'A bi-criteria hybrid Genetic Algorithm with robustness objective for the course timetabling problem', *Computers & Operations Research* **90**, 22–32.

Al-Betar, M. & Khader, A. T. (2012), 'A harmony search algorithm for university course timetabling', *Annals of Operations Research* **194**, 3–31.

Amaral, P. & Pais, T. C. (2016), 'Compromise ratio with weighting functions in a tabu search multi-criteria approach to examination timetabling', *Computers & Operations Research* **72**, 160–174.

Bai, R., Blazewicz, J., Burke, E. K., Kendall, G. & McCollum, B. (2012), 'A simulated annealing hyper-heuristic methodology for flexible decision support', *4OR: A Quarterly Journal of Operations Research* **10**(1), 43–66.

Balanço RAIDES 2019-20 (2019).

Barnhart, C., Bertsimas, D., Caramanis, C. & Fearing, D. (2012), 'Equitable and efficient coordination in traffic flow management', *Transportation Science* **46**(2), 262–280.

Battistutta, M., Ceschia, S., Cesco, F. D., Gaspero, L. D. & Schaerf, A. (2019), 'Modelling and solving the thesis defense timetabling problem', *Journal of the Operational Research Society* **70**(7), 1039–1050.

Beligiannis, G. N., Moschopoulos, C. N., Kaperonis, G. P. & Likothanassis, S. D. (2008), 'Applying evolutionary computation to the school timetabling problem: The Greek case', *Computers & Operations Research* **35**(4), 1265–1280.

Bellio, R., Ceschia, S., Di Gaspero, L., Schaerf, A. & Urli, T. (2016), 'Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem', *Computers & Operations Research* **65**, 83–92.

Braekers, K., Hartl, R. F., Parragh, S. N. & Tricoire, F. (2016), 'A bi-objective home care scheduling problem: Analyzing the trade-off between costs and client inconvenience', *European Journal of Operational Research* **248**(2), 428–443.

Burke, E. K. & Bykov, Y. (2017), 'The late acceptance Hill-Climbing heuristic', *European Journal of Operational Research* **258**(1), 70–78.

Burke, E. K., Marecek, J., Parkes, A. J. & Rudova, H. (2010), 'A supernodal formulation of vertex colouring with applications in course timetabling', *Annals of Operational Research* **179**(1), 105–130.

Burke, E. K., McCollum, B., Meisels, A., Petrovic, S. & Qu, R. (2007), 'A graph-based hyper-heuristic for educational timetabling problems', *European Journal of Operational Research* **176**(1), 177–192.

Calendário Escolar 2019/2020 (2020). Accessed: 2020-09-08.
URL: *https://guiaacademico.tecnico.ulisboa.pt/1o-e-2o-ciclos-e-ciclos-integrados/calendarios/calendario-escolar-20192020/*

Caramia, M., Dell'Olmo, P. & Italiano, G. F. (2008), 'Novel local-search-based approaches to university examination timetabling', *Informs Journal on Computer* **20**(1), 86–99.

Cavdur, F. & Kose, M. (2016), 'A Fuzzy Logic and Binary-Goal Programming-Based Approach for Solving the Exam Timetabling Problem to Create a Balanced-Exam Schedule', *International Journal of Fuzzy Systems* **18**(1), 119–129.

Ceschia, S., Di Gaspero, L. & Schaerf, A. (2012), 'Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem', *Computers & Operations Research* **39**(7), 1615–1624.

Cheong, C. Y., Tan, K. C. & Veeravalli, B. (2009), 'A multi-objective evolutionary algorithm for examination timetabling', *Journal of Scheduling* **12**(2), 121–146.

De Causmaecker, P., Demeester, P. & Vanden Berghe, G. (2009), 'A decomposed metaheuristic approach for a real-world university timetabling problem', *European Journal of Operational Research* **195**(1), 307–318.

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.

Di Gaspero, L., McCollum, B. & Schaerf, A. (2007), The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3), Technical report, Citeseer.

Dung, P., Trung, H., Hoang, T. & Hoang, N. (2015), A java library for constraint-based local search: Application to the master thesis defense timetabling problem, pp. 1–8.

Eley, M. (2007), Ant algorithms for the exam timetabling problem, *in* Burke, EK and Rudova, H, ed., 'Practice and Theory of Automated Timetabling VI', Vol. 3867 of *Lecture Notes in Computer Science*, pp. 364+. 6th International Conference on Practice and Theory of Automated Timeabling, Maasaryk Univ, Brno, CZECH REPUBLIC, AUG 30-SEP 01, 2006.

Elloumi, A., Kamoun, H., Jarboui, B. & Dammak, A. (2014), 'The classroom assignment problem: Complexity, size reduction and heuristics', *Applied Soft Computing* **14**(C), 677–686.

Fastré, L. (2017), Automatic scheduling for master thesis defenses.

Goh, S. L., Kendall, G. & Sabar, N. R. (2017), 'Improved local search approaches to solve the post enrolment course timetabling problem', *European Journal of Operational Research* **261**(1), 17–29.

Huynh, T. T. B., Pham, Q. D. & Pham, D. D. (2012), Genetic algorithm for solving the master thesis timetabling problem with multiple objectives, *in* '2012 Conference on Technologies and Applications of Artificial Intelligence', pp. 74–79.

Jat, S. N. & Yang, S. (2011), 'A hybrid genetic algorithm and tabu search approach for post enrolment course timetabling', *Journal of Scheduling* **14**(6), 617–637.

Kahar, M. N. M. & Kendall, G. (2010), 'The examination timetabling problem at Universiti Malaysia Pahang: Comparison of a constructive heuristic with an existing software solution', *European Journal of Operational Research* **207**(2), 557–565.

Kahar, M. N. M. & Kendall, G. (2015), 'A great deluge algorithm for a real-world examination timetabling problem', *Journal of the Operational Research Society* **66**(1), 116–133.

Kochaniková, B. & Rudová, H. (2013), Student scheduling for bachelor state examinations.

Lach, G. & Luebbecke, M. E. (2012), 'Curriculum based course timetabling: new solutions to Udine benchmark instances', *Annals of Operations Research* **194**(1), 255–272.

Leite, N., Melicio, F. & Rosa, A. C. (2019), 'A fast simulated annealing algorithm for the examination timetabling problem', *Expert Systems with Applications* **122**, 137–151.

Lewis, R., Paechter, B. & McCollum, B. (2007), 'Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition'.

Lewis, R. & Thompson, J. (2015), 'Analysing the effects of solution space connectivity with an effective metaheuristic for the course timetabling problem', *European Journal of Operational Research* **240**(3), 637–648.

Li, J., Burke, E. K. & Qu, R. (2012), 'A pattern recognition based intelligent search method and two assignment problem case studies', *Applied Intelligence* **36**(2), 442–453.

Lü, Z. & Hao, J.-K. (2010), 'Adaptive tabu search for course timetabling', *European Journal of Operational Research* **200**, 235–244.

Master's in Industrial Egineering and Management Dissertations at Instituto Superior Técnico (2020). Accessed: 2020-09-08.
URL: *https://fenix.tecnico.ulisboa.pt/cursos/megi/dissertacoes*

Mavrotas, G. (2009), 'Effective implementation of the ε-constraint method in multi-objective mathematical programming problems', *Applied mathematics and computation* **213**(2), 455–465.

Mavrotas, G. & Florios, K. (2013), 'An improved version of the augmented ε-constraint method (augmecon2) for finding the exact pareto set in multi-objective integer programming problems', *Applied Mathematics and Computation* **219**(18), 9652–9669.

Meseguer, P., Rossi, F. & Schiex, T. (2006), 'Soft constraints', *Handbook of Constraint Programming* pp. 281–328.

Müller, T. & Rudová, H. (2016), 'Real-life curriculum-based timetabling with elective courses and course sections', *Annals of Operations Research* **239**(1), 153–170.

Nothegger, C., Mayer, A., Chwatal, A. & Raidl, G. R. (2012), 'Solving the post enrolment course timetabling problem by ant colony optimization', *Annals of Operations Research* **194**(1), 325–339.

OECD (2018), 'Skills strategy implementation guidance for portugal: Strengthening the adult-learning system', *OECD Skills Studies* .
URL: *http://dx.doi.org/10.1787/9789264298705-en*

Penn, M., Potts, C. & Harper, P. (2017), 'Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables', *European Journal of Operational Research* **262**(1), 194–206.

Phillips, A. E., Waterer, H., Ehrgott, M. & Ryan, D. M. (2015), 'Integer programming methods for large-scale practical classroom assignment problems', *Computers & Operations Research* **53**, 42–53.

Pongcharoen, P., Promtet, W., Yenradee, P. & Hicks, C. (2008), 'Stochastic optimisation timetabling tool for university course scheduling', *International Journal of Production Economics* **112**(2), 903 – 918. Special Section on RFID: Technology, Applications, and Impact on Business Operations.

Qu, R. & Burke, E. K. (2009), 'Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems', *Journal of the Operational Research Society* **60**(9), 1273–1285.

Qu, R., Burke, E. K. & McCollum, B. (2009), 'Adaptive automated construction of hybrid heuristics for exam timetabling and graph colouring problems', *European Journal of Operational Research* **198**(2), 392–404.

Rahimian, E., Akartunalı, K. & Levine, J. (2017), 'A hybrid integer and constraint programming approach to solve nurse rostering problems', *Computers & Operations Research* **82**, 83–94.

Regulamento de Unidades Curriculares de Dissertação (Mestrado) (2020). Accessed: 2020-09-08.
URL: *https://guiaacademico.tecnico.ulisboa.pt/1o-e-2o-ciclos-e-ciclos-integrados/regulamentos/regulamento-de-unidades-curriculares-de-dissertacao/*

Sabar, N. R., Ayob, M., Kendall, G. & Qu, R. (2009), Roulette wheel graph colouring for solving examination timetabling problems, *in* 'International conference on combinatorial optimization and applications', Springer, pp. 463–470.

Sabar, N. R., Ayob, M., Kendall, G. & Qu, R. (2012), 'A honey-bee mating optimization algorithm for educational timetabling problems', *European Journal of Operational Research* **216**(3), 533–543.

Sabar, N. R., Ayob, M., Qu, R. & Kendall, G. (2012), 'A graph coloring constructive hyper-heuristic for examination timetabling problems', *Applied Intelligence* **37**(1), 1–11.

Schimmelpfeng, K. & Helber, S. (2007), 'Application of a real-world university-course timetabling model solved by integer programming', *OR Spectrum* **29**(4), 783–803.

Schrank, D., Eisele, B., Lomax, T. & Bak, J. (2015), '2015 urban mobility scorecard'.

Shi, Y., Boudouh, T. & Grunder, O. (2017), 'A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand', *Expert Systems with Applications* **72**, 160–176.

Soria-Alcaraz, J. A., Ochoa, G., Sotelo-Figeroa, M. A. & Burke, E. K. (2017), 'A methodology for determining an effective subset of heuristics in selection hyper-heuristics', *European Journal of Operational Research* **260**(3), 972–983.

Soria-Alcaraz, J. A., Ochoa, G., Swan, J., Carpio, M., Puga, H. & Burke, E. K. (2014), 'Effective learning hyper-heuristics for the course timetabling problem', *European Journal of Operational Research* **238**(1), 77–86.

Stolletz, R. & Brunner, J. O. (2012), 'Fair optimization of fortnightly physician schedules with flexible shifts', *European Journal of Operational Research* **219**(3), 622–629.

Tangpattanakul, P., Jozefowiez, N. & Lopez, P. (2015), 'A multi-objective local search heuristic for scheduling earth observations taken by an agile satellite', *European Journal of Operational Research* **245**(2), 542–554.

Tramitação da Dissertação/Projeto (Mestrado) (2020). Accessed: 2020-09-08.
URL: *https://guiaacademico.tecnico.ulisboa.pt/1o-e-2o-ciclos-e-ciclos-integrados/regulamentos/tramitacao-da-dissertacaoprojeto/*

Turhan, A. M. & Bilgen, B. (2017), 'Mixed integer programming based heuristics for the patient admission scheduling problem', *Computers Operations Research* **80**, 38–49.

Vermuyten, H., Lemmens, S., Marques, I. & Belien, J. (2016), 'Developing compact course timetables with optimized student flows', *European Journal of Operational Research* **251**(2), 651–661.

Wang, S., Bussieck, M., Guignard, M., Meeraus, A. & O'Brien, F. (2010), 'Term-end exam scheduling at united states military academy/west point', *Journal of Scheduling* **13**, 375–391.

Woumans, G., De Boeck, L., Belien, J. & Creemers, S. (2016), 'A column generation approach for solving the examination-timetabling problem', *European Journal of Operational Research* **253**(1), 178–194.

Yang, S. & Jat, S. N. (2011), 'Genetic algorithms with guided and local search strategies for university course timetabling', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **41**(1), 93–106.

Zhang, L., Bouchet, P.-Y., Lamine, E., Fontanili, F., Bortolaso, C., Derras, M. & Pingaud, H. (2019), Home health care daily planning considering the satisfaction of all the stakeholders, *in* ‘2019 International Conference on Industrial Engineering and Systems Management (IESM)’, IEEE, pp. 1–6.

Öztürké, M., Tsoukiàs, A., Vincke, P., Figueira, J., Greco, S. & Ehrogott, M. (2005), *Preference Modelling*, pp. 27–59.

Appendix A

User Guide

For both approaches, a tool to make it possible for someone without Python knowledge to schedule thesis defences was implemented. In this chapter, the necessary steps to use this tool are explained.

A.1 Required Software

Both approaches were implemented using Python 3.7 and to use the tool this is a requirement. Moreover, the python libraries pandas, glob and datetime are also necessary. Furthermore, a Gurobi license is needed. Lastly, since the Python tool will import all the data from an Excel file and write the schedules in another one, Excel is also necessary.

A.2 Two Stage *a Priori* Approach

The required data is specified in Table 4.1 to which the weights for each of the three objectives are added. All of these are provided through an Excel file named “Dados.xlsx” with the exception of the committee members time-slot preferences, for whom there are Excel files named after each committee member identification number.

In the Excel file “Dados.xlsx”, four sheets can be found. The first one is named “Parâmetros” and it can be seen in Figure A.1. In the first column the number of days between the first and last day should be clarified, in the second column the number of quarter-hours in a day where a thesis defence can be scheduled, in the third the number of available rooms, in the fourth the duration of a thesis defence in quarter-hours, in the fifth the quarter-hours that are considered the compact zone and in the last two columns the hours and minutes of the first available hour each day.

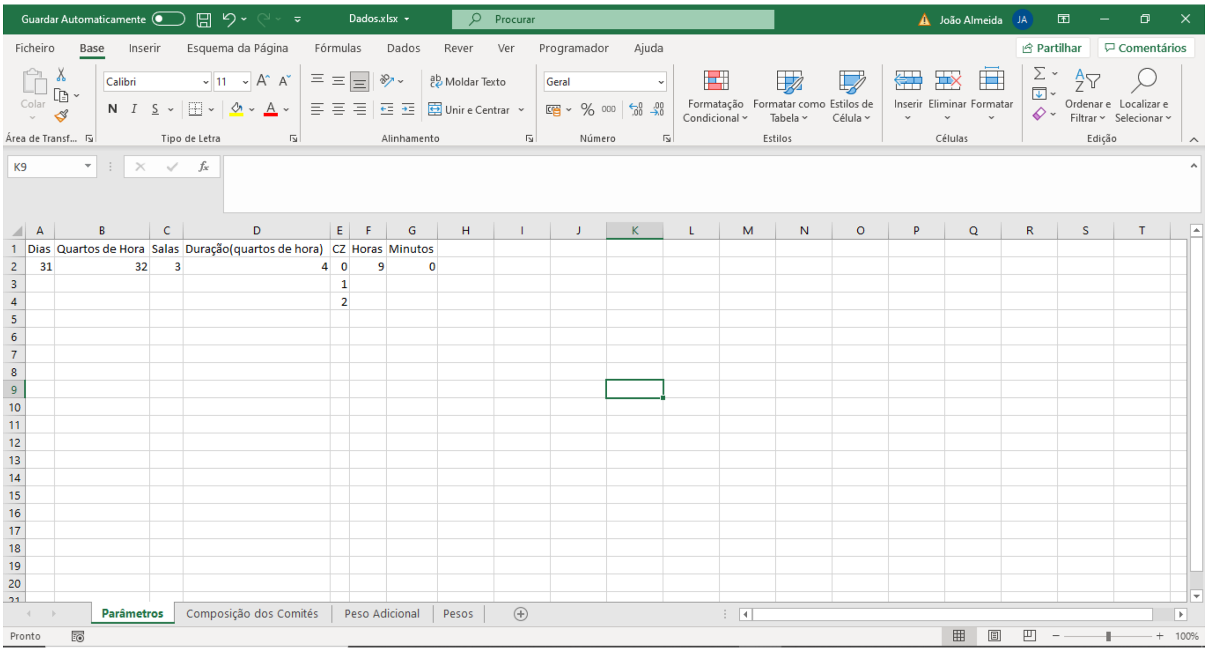


Figure A.1: Sheet Parâmetros

The second sheet is named “Composição dos Comités” and an example can be seen in Figure A.2. This sheet is where the committee composition for every thesis defence should be written. In the first column the student number of the defendant should be included and in the following three columns the identification numbers of the chairperson, supervisor and additional member, respectively.

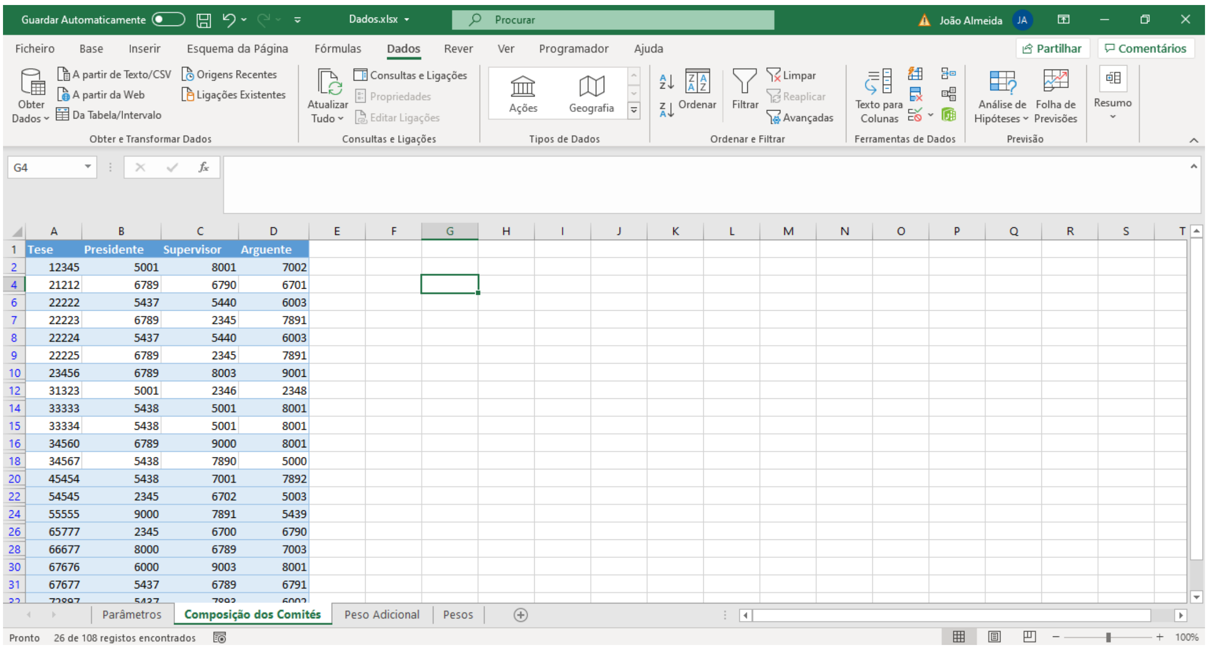


Figure A.2: Sheet Composição dos Comités

The third sheet is named “Peso Adicional” and it is represented in Figure A.3. In it, the decision-maker can indicate the members that should be awarded the additional scheduling weight O.

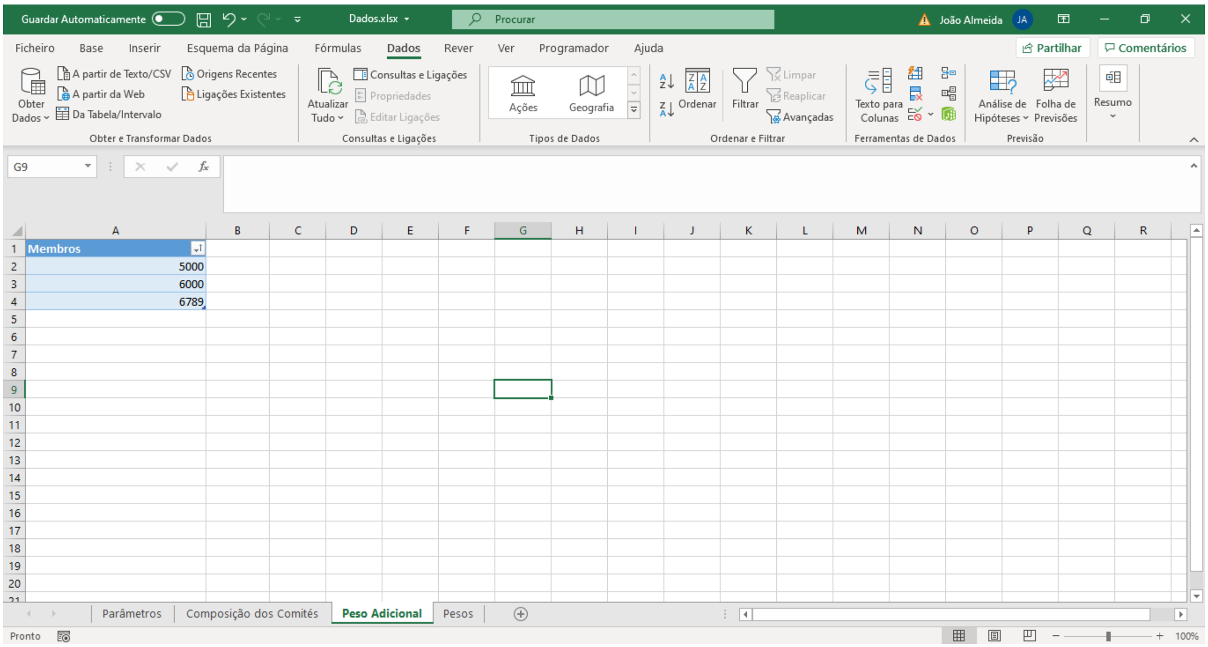


Figure A.3: Sheet Peso Adicional

In the last sheet named “Pesos”, represented in Figure A.4, the decision-maker should indicate the several weights that are in the model. Specifically, the first column corresponds to the weight for distinguishing chairpersons, the second column the additional weight, the third column the weights for each of the zones considered compact, the fourth column the weight referring to the number of days a committee member is scheduled for, the fifth column the weight regarding the time-slot preference objective and, lastly, the sixth column the weight regarding the compactness objective.

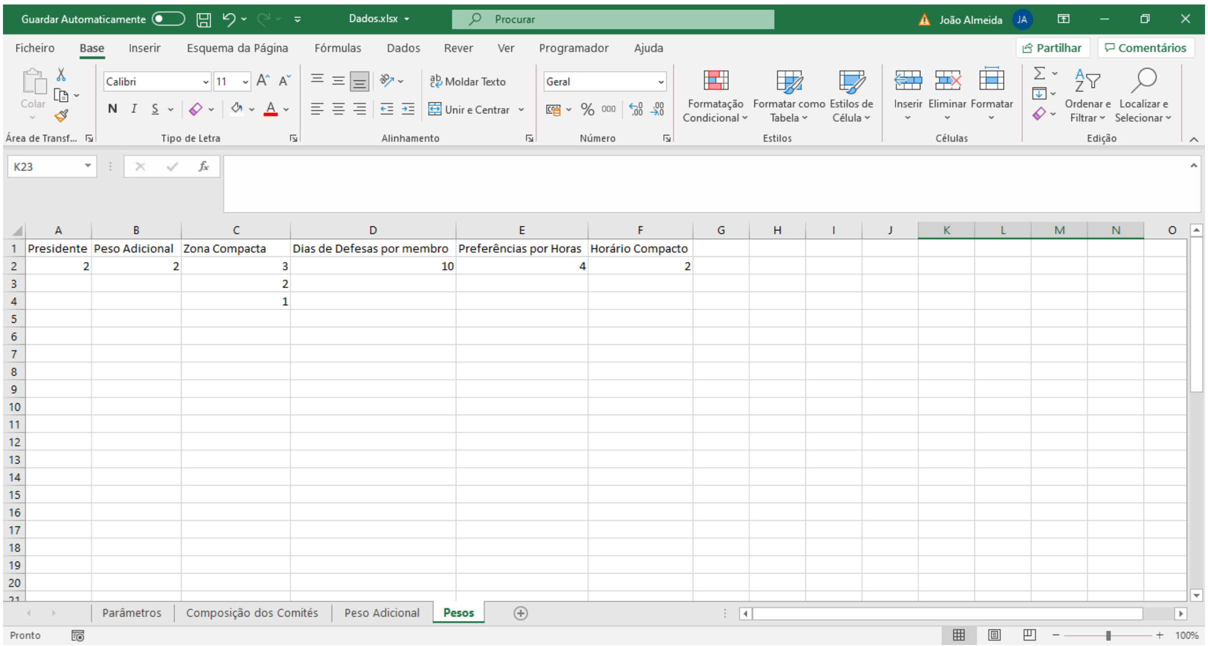


Figure A.4: Sheet Pesos

Finally, the only data that is missing is the committee members availability. To store it, a folder named schedules was created. In it, there will be an Excel file for every committee member named after their identification number.

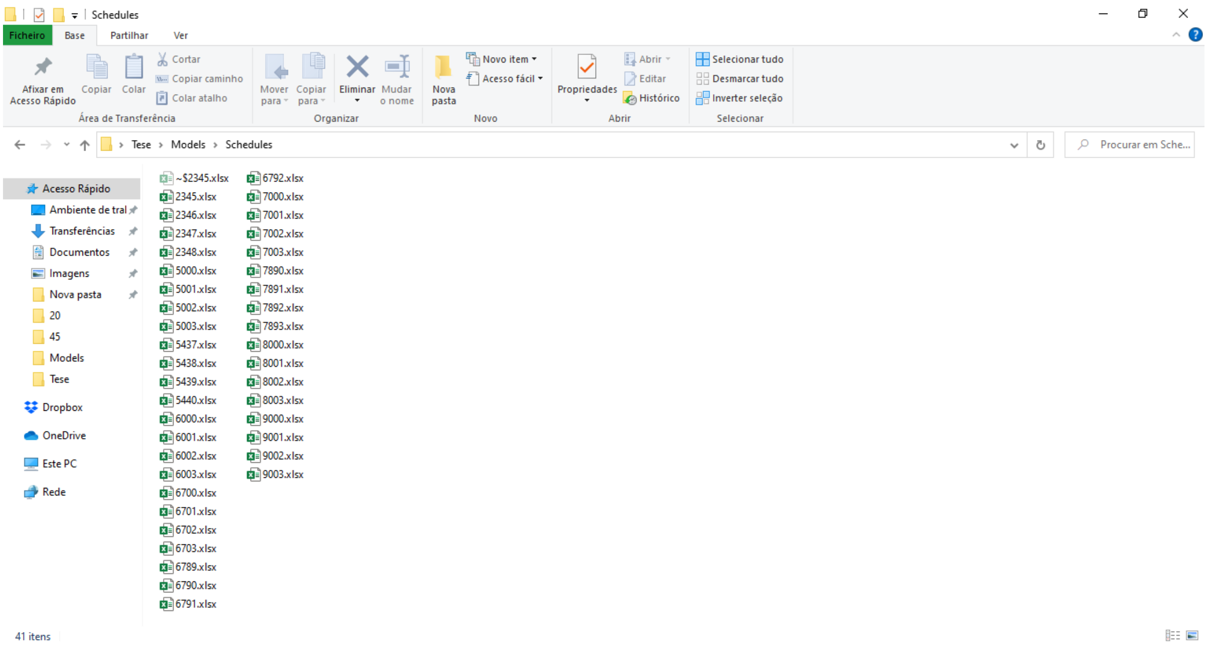
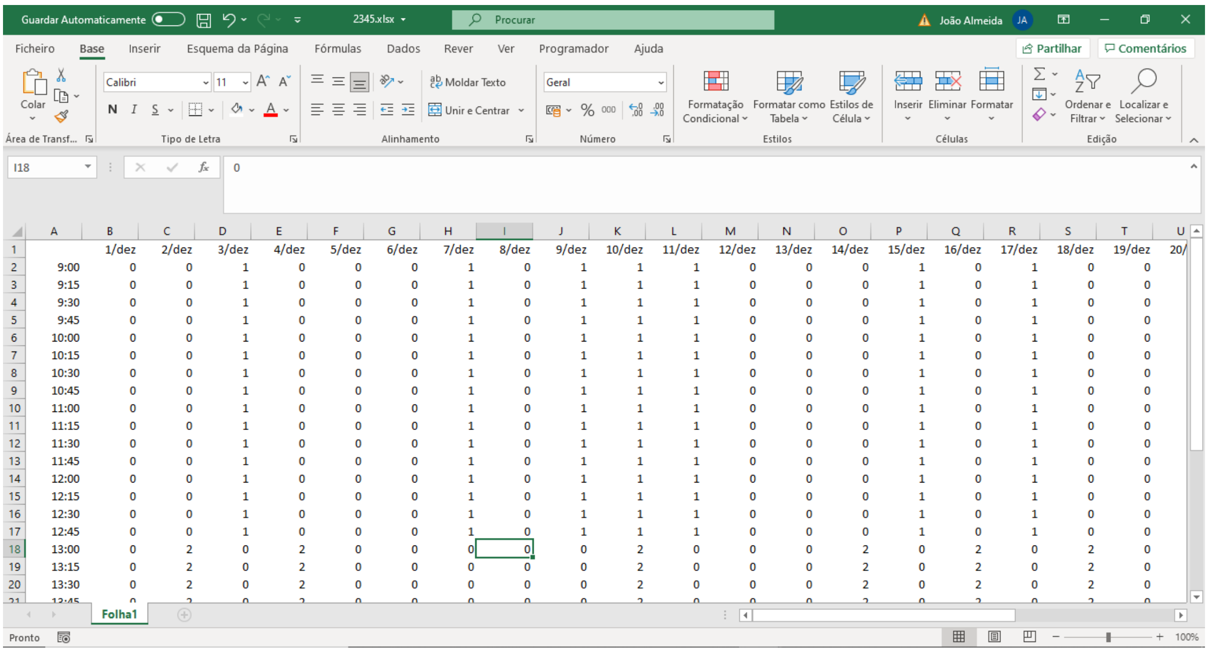


Figure A.5: Folder Schedules

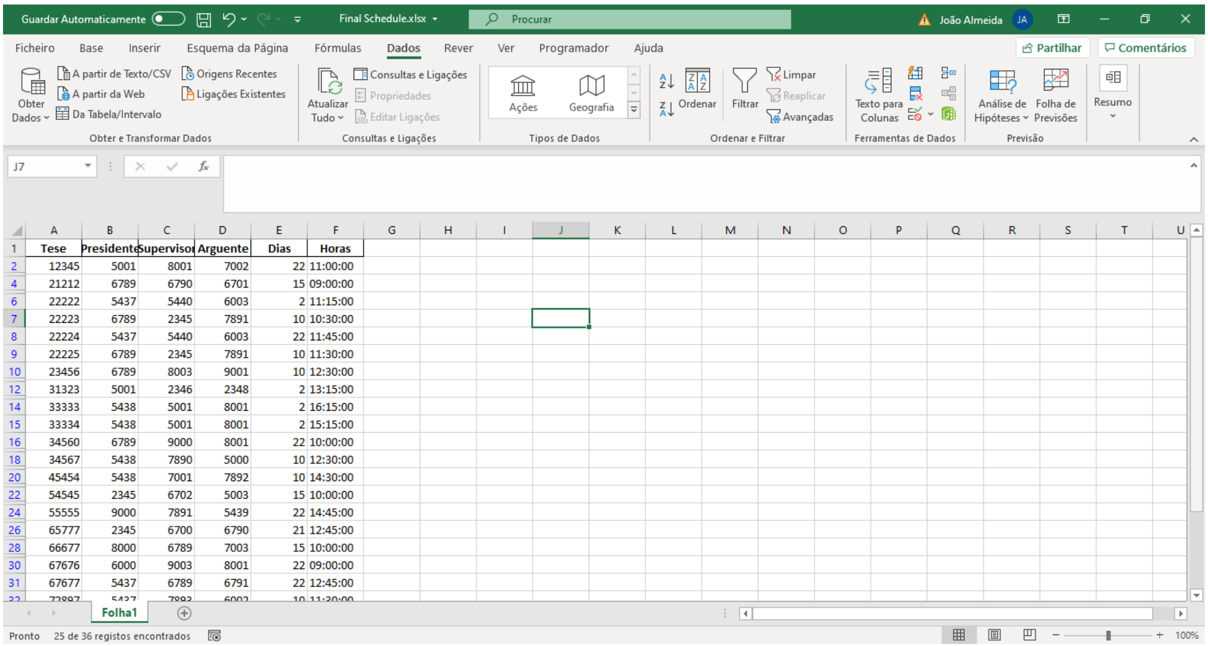
Then, in every file, the preferences can be submitted for the corresponding committee member, with a 0 for a time-slot translating into an unavailable time-slot for a defence to start at and a positive integer to an available time-slot. The larger that number the higher the level of preference for that slot.



	1/dez	2/dez	3/dez	4/dez	5/dez	6/dez	7/dez	8/dez	9/dez	10/dez	11/dez	12/dez	13/dez	14/dez	15/dez	16/dez	17/dez	18/dez	19/dez
9:00	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
9:15	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
9:30	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
9:45	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
10:00	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
10:15	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
10:30	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
10:45	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
11:00	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
11:15	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
11:30	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
11:45	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
12:00	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
12:15	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
12:30	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
12:45	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	0	0
13:00	0	2	0	2	0	0	0	0	0	2	0	0	0	2	0	2	0	2	0
13:15	0	2	0	2	0	0	0	0	0	2	0	0	0	2	0	2	0	2	0
13:30	0	2	0	2	0	0	0	0	0	2	0	0	0	2	0	2	0	2	0

Figure A.6: Example of a Time-Slot Preference File

It is now possible to run the Python file, which will import all the data and find the optimal solution. This solution will then be exported to another Excel file named "Final Schedule.xlsx", for which an example can be seen in Figure A.7.



	Tese	Presidente	Supervisor	Arguente	Dias	Horas
2	12345	5001	8001	7002	22	11:00:00
4	21212	6789	6790	6701	15	09:00:00
6	22222	5437	5440	6003	2	11:15:00
7	22223	6789	2345	7891	10	10:30:00
8	22224	5437	5440	6003	22	11:45:00
9	22225	6789	2345	7891	10	11:30:00
10	23456	6789	8003	9001	10	12:30:00
12	31323	5001	2346	2348	2	13:15:00
14	33333	5438	5001	8001	2	16:15:00
15	33334	5438	5001	8001	2	15:15:00
16	34560	6789	9000	8001	22	10:00:00
18	34567	5438	7890	5000	10	12:30:00
20	45454	5438	7001	7892	10	14:30:00
22	54545	2345	6702	5003	15	10:00:00
24	55555	9000	7891	5439	22	14:45:00
26	65777	2345	6700	6790	21	12:45:00
28	66677	8000	6789	7003	15	10:00:00
30	67676	6000	9003	8001	22	09:00:00
31	67677	5437	6789	6791	22	12:45:00

Figure A.7: Final Schedule

A.3 Two Stage Augmented ϵ - Constraint Approach

While there are some differences between the data necessary to run this approach, the way the data is imported is the same, with some adjustments to the Excel files where it is written. For starters, the steps to import the data regarding the time-slots preferences are exactly the same, including the folder Schedules and the filling of each Excel file for each committee member. On the other hand, the other file which was previously named “Dados.xlsx” is no longer used, with a different one named “Dados E-Const.xlsx” substituting it.

Given the similarities between both approaches, the three first sheets in both Excel files are the same, with the only difference being found in the fourth sheet where the weights are indicated. Since in the AUGMECON approach there are no objective weights,these are no longer needed, however, the value for epsilon must be given. The sheet “Pesos” for this approach is presented in Figure A.8.

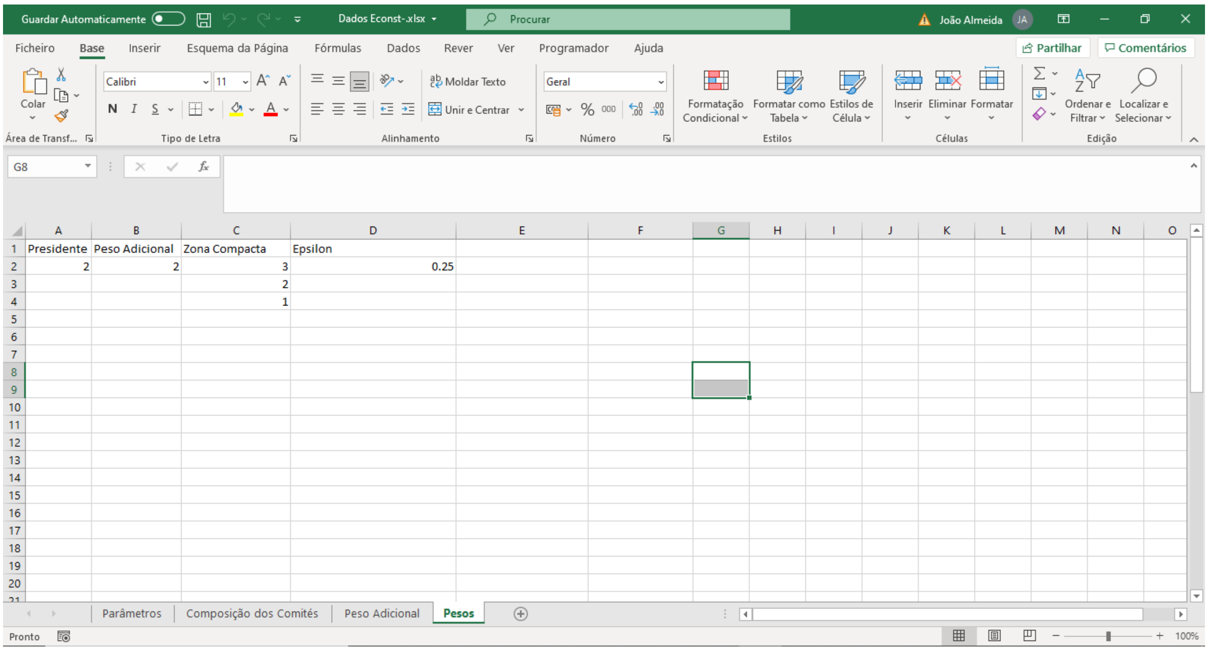


Figure A.8: Sheet Pesos for the AUGMECON approach

Furthermore, the method for presenting the results is the same as in the previous approach, with the difference that instead of only one schedule possibility being presented, all of the generated schedules are presented instead.