# An Integer Programming Model for Scheduling Master's Thesis Defences

João António Carvalho Almeida

Instituto Superior Técnico

**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 $\epsilon$ - 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 the Augmented $\epsilon$ - 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 $\epsilon$ - 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:** Master Thesis Defence Scheduling, Preference Modelling, Fairness, Integer Programming, Multi-objective, $\epsilon$ - constraint method

## 1 Introduction

Instituto Superior Técnico (IST), as part of the University of Lisbon, must adhere to the country's regulations regarding second study cycle degree courses and dissertation discussions. These are all currently described in (1), which has been emitted after an assessment of the Portuguese higher education system, requested by the government to the Organisation for Economic Co-operation and Development.

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 dissertation in the institution. 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 dissertation 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 Department of Engineering and Management (DEG) at IST's processes for scheduling master thesis defences, which oversees both research work and the bachelor's (LEGI) and master's (MEGI) Degree in Industrial Engineering and Management at ist, as well as a few other doctorate degree courses.

In the department in question, the secretary oversees the scheduling of all the dissertation 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 a different *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.

Each year, two different deadlines are set for submitting dissertations. Consequently, during that time, there is an overload of work to the department's secretary, with up to 34 thesis defences having had to be scheduled in a single month in 2019. To define priorities, there is usually a preference given to the chairpersons, who usually have more defences to be present at, 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.

Thesis defence scheduling is a recent branch within the academic scheduling field which has recently been receiving increased attention. However, 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.

Accordingly, the aim of this thesis is to automate this process at the DEG at IST 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 ist.

## 2 Literature Review

While the literature on other Academic Scheduling problems is quite vast and well-developed, there are only five other works on the thesis scheduling problem as far as the author is aware.

A few different approaches have been taken to tackle this problem, namely genetic algorithm (8), local search (9; 6), optimization models (7; 3) and simulated annealing (3).

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 (9; 7) or five (8; 6), with their respective functions. Nonetheless, in the case of the University of Udine (3), 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 (8; 9; 6; 7), however, there are also cases where some members of the examination committee are assigned to complete sessions of several defences instead (9; 7) and one case where all the examination committee members are assigned through this method (3).

The most common objectives are reducing or evening out the total workload of the professors (9; 6; 7; 3) and creating more compact schedules for the examination committee members (8; 9; 7). Furthermore, the suitability between the theme of a thesis and the examination committee assigned to it is considered in some of them (8; 6; 3), as well as reducing the changes in the classrooms(8) and adding a reserve day when defences should not be scheduled (7).

The number of students varies between the cases, ranging from a minimum of 9 (8; 6) to a maximum of 551 (3). 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 (3) 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 (3) 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.

A comparison between the previous literature and this work is presented in Table 1.

Table 1: Summary of the Literature on Thesis Defence Scheduling Problems

| | **Solution Method** | | | | **Examination Committe Assignment** | | **Objectives** | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Paper** | GA | LS | IP | SA | Preset | Mixed | 1) | 2) | 3) | 4) | 5) | 6) | 7) |
| (8) | ✓ | | | | | ✓ | | ✓ | ✓ | | ✓ | | ✓ |
| (9) | | ✓ | | | | ✓ | ✓ | | ✓ | | | | |
| (6) | | ✓ | | | | ✓ | ✓ | ✓ | | | | | |
| (7) | | | ✓ | | | ✓ | | | ✓ | ✓ | | | ✓ |
| (3) | | | ✓ | ✓ | | ✓ | ✓ | ✓ | | | | | |
| Present Work | | | ✓ | | ✓ | | | | ✓ | | | ✓ | ✓ |

**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

Given the characteristics of the case at hand, a review on the literature on important concepts such as preference modelling and scheduling for fairness was also conducted. With other works from the other academic scheduling problems and other unrelated scheduling fields being analysed.

In course timetabling, introducing individual preferences is not uncommon, with works such as (18; 15; 14) and (23) including professors individual requests in their formulations. Furthermore, in healthcare related scheduling problems this concept also occurs. For example, in home health care problems where the goal is to optimize the operational costs of the companies (4; 19; 24), patient admission problems which aim to assign patients to hospital beds (22), nurse rostering problems with the objective of allotting shifts to nurses (16) and operating theatre timetabling problems which seek to schedule surgeries (13).

Similarly to most cases in this section, the thesis defence scheduling case at IST combines general rules and individual requests, thus it becomes necessary to find solution approaches that take this into consideration, given the multi-objective nature of the problem. Furthermore, there were some interesting concepts that might be of use, such as distinction between different staff regarding the positions they occupy (23; 24), incremental penalties for cases where one user is assigned several least preferred time slots (13) or the penalization of requests that are deemed detrimental to other users if they were to be complied with (18).

Ensuring fairness amongst all schedules for each committee member is an almost impossible task, still there are mechanisms that can work towards such a goal. Besides papers from academic scheduling problems (17; 9; 6; 23; 7; 5; 3; 10), other areas that focus on this goal were also reviewed, namely are healthcare (20), traffic flow management in an airport (2) and scheduling of earth observations made by a satellite (21).

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 (20; 9; 6; 3), the spread of events (17; 5; 10), the assignment of undesirable time slots (13) or the profit each individual can take from the scheduling (21). 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 (2), position of the professional (23) or the number of predetermined events someone must attend (7).

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.

# 3 Mathematical Model

## 3.1 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, the supervisors and an additional member. 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.´

## 3.2 Problem Formulation

The problem was modelled as a mixed integer linear programming model, with multiple objectives. Let T denote the set of thesis defences that need to be scheduled. These thesis defences 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.

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 MWW. 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 OW, 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 minimizing 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.

### 3.2.1 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 (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 (2). Furthermore, it is also necessary to guarantee that committee members have no juxtaposed defences (3). Finally, there is a constraint that ensures that there are no more thesis defences at a time than rooms available (4).

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

$$X_{tdh} \leq HP_{CM_{tp}dh} \;\; \forall \;\; t \in T, p \in P, d \in D, h \in H \tag{2}$$

$$X_{tdh} + \sum_{l=0}^{L-1} X_{t1,d,h-l} \leq 1 \;\; \forall \;\; t \in T, t1 \in T, d \in D, h \in H, t \neq t1, ICM_{t,t1} = 1 \tag{3}$$

$$\sum_{t=0}^{T} \sum_{l=0}^{L-1} X_{t,d,h-l} \leq R \;\; \forall \;\; d \in D, h \in H \tag{4}$$

### 3.2.2 Taguspark *Campus* Presence Constraints

The Taguspark *Campus* presence constraints (5)–(10) set the values for variables $Y_{md}$, $G_m$ and $GQ_{mq}$.

The first two constraints, (5) and (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. $G_m$ is an integer variable that represents the number of days committee member m has discussions scheduled on (7), furthermore, it is necessary to its upper bound (8). To finish, the value of $GQ_{mq}$ is set by two constraints, (9) and (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.

$$\sum_{t1=0}^{T} \sum_{h=0}^{H} X_{t1,d,h} \geq Y_{CM_{tp}d} \;\; \forall \;\; t \in T, p \in P, d \in D, ICM_{t,t1} = 1 \tag{5}$$

$$X_{tdh} \leq Y_{CM_{tp}d} \;\; \forall \;\; t \in T, p \in P, d \in D, h \in H \tag{6}$$

$$\sum_{d=0}^{D} Y_{md} = G_m \;\; \forall \;\; m \in M \tag{7}$$

$$G_m \leq B \;\; \forall \;\; m \in M \tag{8}$$

$$\sum_{q=0}^{Q} GQ_{mq} = 1 \;\; \forall \;\; m \in M \tag{9}$$

$$\sum_{q=0}^{Q} GQ_{mq} \times q = G_m \;\; \forall \;\; m \in M \tag{10}$$

### 3.2.3 Compactness Constraints

The compactness constraints, (11)–(15), are intended to provide a measure of schedule compactness by setting the values for variables $C_{tdh}$ and $Z_{tdh}$.

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 CZ 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 (11), with $C_{tdh}$ being the integer variable that represents this measure.

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. Thus, a big-M approach was taken, and implemented in constraints (12)–(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.

$$C_{tdh} = \sum_{t1=0}^{T}\sum_{m=0}^{M}\sum_{cz=0}^{CZ} MW_m \times O_m \times CZW_{cz} \times X_{t1,d,h-L-cz} \;\; \forall \;\; t \in T, d \in D, h \wedge h-L-cz \in H, cz \in CZ, m \in CM_t \wedge CM_{t1}, t \neq t1 \tag{11}$$

$$Z_{tdh} \geq 0 \;\; \forall \;\; t \in T, d \in D, h \in H \tag{12}$$

$$Z_{tdh} \leq C_{tdh} \;\; \forall \;\; t \in T, d \in D, h \in H \tag{13}$$

$$Z_{tdh} \leq BM \times X_{tdh} \;\; \forall \;\; t \in T, d \in D, h \in H \tag{14}$$

$$Z_{tdh} \geq C_{tdh} - BM \times (1 - X_{tdh}) \;\; \forall \;\; t \in T, d \in D, h \in H \tag{15}$$

### 3.2.4 Objectives

For the case at hand, four different objectives to be maximised were considered. The first objective (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.

$$Max \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} X_{tdh} \tag{16}$$

The second objective (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.

$$Max \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \tag{17}$$

The third objective (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.

$$Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} \tag{18}$$

Lastly, as was previously explained, the fourth objective (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 (17; 18).

$$Max \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \tag{19}$$

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 (17), (18) and (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 Solution Approach

## 4.1 Two Stage Multi-objective Optimisation Strategy

For the case at hand, the objective of maximising the number of thesis defences scheduled (16) can be clearly identified as the primary one. Regardless of the values of the other three objectives (17)–(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 (20).

$$\sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} X_{tdh} = TB \tag{20}$$

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 $\epsilon$-constraint based approach, based on the works (11) and (12) and which the authors name AUGMECON.

## 4.2 *a Priori* Optimisation Approach - Second Stage

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 (21) that can be maximised without any additional considerations.

$$\begin{aligned} &Max \\ &HPW \times \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \\ &\quad - GW \times \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} \\ &\quad + ZW \times \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \end{aligned} \tag{21}$$

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

- $HPW$ - Weight regarding the objective of maximizing the preferences of each member regarding time slots (17);
- $GW$ - Weight regarding the objective of minimizing the number of days a member is scheduled on (18);
- $ZW$ - Weight regarding the compactness of schedules objective (19)

## 4.3 Augmented $\epsilon$ - Constraint Approach - Second Stage

For further understanding of the employed approach, the reader is deferred to the works (11) and (12).

To implement the AUGMECON algorithm to the MEGI thesis defence problem, several new parameters and variables are necessary.

As both objectives (17) and (19) are the ones being bounded, the main objective to be maximised is the objective (18). The positions from each objective are interchangeable.

First, the $\epsilon$ parameter must be defined. The algorithm was implemented considering a value greater than 0 and lower than or equal to 0.5. 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, 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.

After the first stage, 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 (17) and (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 (17) or (19) was to set the value for the opposite objective as its maximum and then optimize the model considering as the objective function objective (18) plus either (17) or (19) multiplied by an adequately small number, epss, which, for the case at hand, was set as $10^{-4}$. Thus, to find NP, constraint (23) is added to the model and objective (22) is maximised and, in turn, to find NZ, constraint (25) is added to the model and objective (24) is maximised.

$$\begin{aligned} &Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} + epss \\ &\times \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \end{aligned} \tag{22}$$

$$\sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} = MZ \tag{23}$$

$$\begin{aligned} &Max - \sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} + epss \\ &\qquad \times \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \end{aligned} \tag{24}$$

$$\sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} = MP \tag{25}$$

Before starting to search for solutions, it is also necessary to define the slack variables sp (constraint 26) and sz (constraint 27). 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.

$$sp = MP - \sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \tag{26}$$

$$sz = MZ - \sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \tag{27}$$

The objective function 28 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 (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.

$$-\sum_{m=0}^{M}\sum_{q=0}^{Q} O_m \times MW_m \times GQ_{mq} \times q^{EXP} + eps \times (\frac{sz}{MZ} + \frac{sp}{MP}) \tag{28}$$

The method also has a specific mechanism to iterate between different solutions. Just like the conventional $\epsilon$ - constraint method, this mechanism is based on setting different combinations of lower bounds for the objectives, which, for the case at hand, are objectives (17) and (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:

$$\sum_{t=0}^{T}\sum_{p=0}^{P}\sum_{d=0}^{D}\sum_{h=0}^{H} O_{CM_{tp}} \times MW_{CM_{tp}} \times HP_{CM_{tp}dh} \times X_{tdh} \geq LBP \tag{29}$$

$$\sum_{t=0}^{T}\sum_{d=0}^{D}\sum_{h=0}^{H} Z_{tdh} \geq LBZ \tag{30}$$

Moreover, what guarantees that different solutions are searched is setting different values for LBP and LBZ in each iteration. This is achieved by adding $\epsilon$ times the difference between the maximum value of an objective and its worst value every time a new iteration starts. 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. It is possible to foresee the maximum number of solutions the algorithm can find based on the value of $\epsilon$ 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.

# 5 Computational Experiments

All computational experiments were conducted using a Intel(R) Core(TM) i7-6500 CPU @ 2.50GHz2.59GHz and 8GB of installed RAM. Moreover, the employed software was Python 3.7 and Gurobi 9.0.0.

All instances used were randomly generated.

## 5.1 Instance Generator

Three different parameters that need to be generated were 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.

## 5.2 Instances

All the instances were created for a time period of 20 days, as well as 32 quarter-hours available each day for thesis defences to start. Three major groups of instances were created, the first one (A) corresponding to instances with 30 defences, the second (B) 40 defences and the last one (C) with 50 defences. Moreover, within each of them, three other subgroups were created, with the first one (1) having a percentage of available time for committee members of 20 percent, the second (2) 45 percent ant the last (3) 70 percent. Furthermore, for each of these instance subgroups, three instances were randomly generated.

## 5.3 *a Priori* Two Stage Approach

### 5.3.1 First Stage

Out of the 27 instances tested, in 23 it was possible to schedule the entirety of the discussions, proving the usefulness of not setting the number of thesis defences to be scheduled as an hard constraint from the start. Furthermore, in every instance where this was not possible only one of the thesis defences remained unscheduled. Every first stage optimisation took less than 1 minute.

#### 5.3.2 Second Stage

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.

#### 5.3.3 Number of Rooms

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 (18). It was 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.

#### 5.3.4 Compactness Constraints

The compactness constraints were formulated with a resource to the big-M method. 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 (36) 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 2: Summary of the Base Instances Results

| | | 1$^{st}$ Stage | | 2$^{nd}$ Stage | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Group** | **Instance** | Scheduled Defences | Time (s) | G | HP | Z | Obj Function | Time (s) | Gap |
| A1 | avg | 29.3 | 4.7 | -319 | 236 | 10.7 | -576 | 4.3 | 0 |
| A2 | avg | 30 | 3.3 | -125.3 | 240.3 | 48.7 | 320 | 135.3 | 0 |
| A3 | avg | 30 | 19.7 | -54.7 | 258 | 66.3 | 688 | 8047.3 | 1.6 |
| B1 | avg | 40 | 16 | -477.3 | 299 | 9.7 | -993 | 5.7 | 0 |
| B2 | avg | 40 | 14 | -157.3 | 330.3 | 60.3 | 482.3 | 338.3 | 0 |
| B3 | avg | 40 | 42 | -90 | 326.3 | 84 | 787 | 10800 | 15.9 |
| C1 | avg | 49.3 | 20 | -601.7 | 389.3 | 21.7 | -1195.3 | 7 | 0 |
| C2 | avg | 50 | 30.3 | -201.7 | 401.7 | 77.3 | 553 | 3272.3 | 0 |
| C3 | avg | 50 | 49 | -120 | 413.7 | 104 | 969 | 10800 | 18.3 |

### 5.4 Two Stage Augmented $\epsilon$ - Constraint Approach

In this approach only instances from groups A1 and A2 were tested. This was the choice due to the approach in question not being as efficient as the first one, as it must iterate between several different lower bounds and solve the model numerous times, until it reaches its conclusion.

#### 5.4.1 Second Stage Initialisation

While all the points (MP, NP, MZ and NZ) 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.

#### 5.4.2 Skipped Iterations and Number of Solutions

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 timesslots 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.

### 5.4.3 Time to Solve each Iteration

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.

A defined trend can be found in the behaviour of the time it takes to solve each iteration. As previously explained, the proposed augmented $\epsilon$ - 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 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.

Table 3: Average Time per Effective Iteration and Estimated Time Gained Through Skipping Iterations

| | | **Effective Iteration Time** | | | | **Estimated Time Gained** | |
|---|---|---|---|---|---|---|---|
| | | **Solutions** | | **Infeasible** | | | |
| **Group** | **Instance** | Average | Total | Average | Total | Feasible | Infeasible |
| $\epsilon = \frac{1}{6}$ Iterations = 49 | | | | | | | |
| A1 | avg | 4.3 | 72 | 4.4 | 13.7 | 5.9 | 124.5 |
| A2 | avg | 369.7 | 8073.3 | 2116.2 | 9002.7 | 848.4 | 45983.7 |
| $\epsilon = \frac{1}{4}$ Iterations = 25 | | | | | | | |
| A1 | avg | 4 | 40 | 4 | 10.7 | 4 | 45.3 |
| A2 | avg | 608.3 | 7757.3 | 855 | 2565 | 1038 | 7359.3 |
| $\epsilon = \frac{1}{2}$ Iterations = 9 | | | | | | | |
| A1 | avg | 3.9 | 18 | 4 | 6.7 | 0 | 11.7 |
| A2 | avg | 562 | 3277 | 111.3 | 222.7 | 0 | 209.3 |

# 6 User Guide

To make both solutions available to the decision-maker, they were implemented using the Python language together with the commercially available solver Gurobi.

Furthermore, all the necessary inputs to the models, including sets and parameters, can be introduced in a set of Excel sheets, also made available to the decision-maker, which the python application will then import. Moreover, not all of these inputs have to be written, as some of them can be inferred from the values of other ones, for example the table ICM or the set Q, which are automatically created based on other information.

Just like the inputs, the outputs are also given through another Excel sheet, in a framework that is easier to read than the one from the python application.

# 7 Conclusions and Future Work

The scheduling of thesis defences in the Department of Engineering and Management at Instituto Superior Técnico is the responsibility of the department's secretary. While there is already some literature on this topic, it is a relatively recent subject of interest within the Academic Scheduling field.

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 variables and constraints.

Four different objectives were considered. The first one is the minimisation of the thesis defences that would go unscheduled. The second objective aims to give the committee members the liberty of giving different preference levels to their available time-slots. 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*. Lastly, the final objective was to ensure that within the same day, the schedule for a certain committee member is as compact as possible.

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. 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. The second proposed method was named Two Stage Augmented $\epsilon$ - Constraint Approach which is based on the $\epsilon$ - constraint method, with the Two Stage *a Priori* Approach being considerably faster in its resolution than the Augmented $\epsilon$ - Constraint Approach, as it will only solve the model twice, whereas the second will have multiple iterations, depending on the value for the $\epsilon$, on the other hand, the Augmented $\epsilon$ - Constraint Approach provides a more complete notion of the solution space and suggests several possible solutions for the decision-maker to chose 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 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.

The effect of varying several parameters was also tested, with some of the 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.

Due to the Covid-19 pandemic and other restrictions it was not possible to apply either approach to real world instances. Thus, the main goal going forward would be to test both approaches using a real world instance and organise 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.

# References


[1] Decreto-lei n.º 65/2018 de 16 de agosto do ministério da ciência, tecnologia e ensino superior. Diário da República n.º 157/2018, Série I, 2018.

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

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

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

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

[6] Dung, P., Trung, H., Hoang, T., and Hoang, N. A java library for constraint-based local search: Application to the master thesis defense timetabling problem. pp. 1–8.

[7] Fastré, L. Automatic scheduling for master thesis defenses.

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

[9] Kochaniková, B., and Rudová, H. Student scheduling for bachelor state examinations.

[10] Leite, N., Melicio, F., and Rosa, A. C. A fast simulated annealing algorithm for the examination timetabling problem. *Expert Systems with Applications 122* (MAY 15 2019), 137–151.

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

[12] Mavrotas, G., and Florios, K. 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 (2013), 9652–9669.

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

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

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

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

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

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

[19] Shi, Y., Boudouh, T., and Grunder, O. A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand. *Expert Systems with Applications 72* (2017), 160–176.

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

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

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

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

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