A CONSTRUCTIVE TOOL TO PREDICT TIMETABLE FEASIBILITY UNDER USER DEFINED CONSTRAINTS.

MASTER THESIS INDUSTRIAL ENGINEERING & MANAGEMENT

12-02-2016

Author:	Stef van den Bedem BSc
	s0192880
Supervisors:	dr. P.C. Schuur
	ir. H. Kroon
	R.W. Puyt MSc MMC
	R.A. Kouthoofd MSc
Programme:	Industrial Engineering & Management
Track:	Production & Logistic Management

SUMMARY

This thesis forms the completion of the master Industrial Engineering & Management at the University of Twente with a specialisation in Production & Logistic Management. The thesis describes the development of a tool to predict the feasibility of timetables under varying constraints. This tool is created for the management of the Business Economics programme at the Amsterdam University of Applied Sciences (AUAS).

APPLICATION

The tool can be used to predict whether or not feasible timetables can be used under pre-specified constraints. These constraints consist of limited resources like lecture rooms and teaching staff and scheduling rules like "lecturers may not work more than ten hours per day".

By scheduling a set of events (lectures, tutorials, etc.) repeatedly in a decreasing number of rooms, we can identify the point at which the pool of rooms becomes too small to include all events in a timetable. With this information we can predict whether or not a feasible timetable can be found under the given constraints or how much space is needed to ensure that a feasible timetable can be created.

Questions that can be addressed include:

- Would introducing new policies create timetabling difficulties, given the current resources?
- Which scheduling constraints can be used simultaneously and which combinations have negative influence on timetable feasibility?
- What are the minimum resource levels given the current set of events and policies?
- What the impact of policies on timetabling?

MOTIVATION

The AUAS is a budget driven organisation with as one of its goals preparing students for a professional career in a wide range of fields. I.e. it tries to offer the best education possible within a set budget.

The AUAS is split in various faculties that each coordinate the research and education in their own field. The faculties have organised the education in a number of programmes. Each faculty has also an operational branch. The programme management decides what to teach, when to teach, how to teach and by whom courses are taught. The space used for teaching on the other hand is managed by the operations bureau. Which leaves the timetabling department with the task to fit all the events given by the educational branch in the set of rooms supplied by the operational branch.

Timetabling is the point where all interests collide and results in a compromise. All parties have reasons to change the compromise: the operations bureau can improve its financial position by reducing space usage; timetablers can produce better timetables when the availability of teaching staff is increased; whereas more rooms would enable programme managers to increase student attendance by only using favourable lecture hours.

However, a lot of related questions are difficult to answer, like: What is the absolute minimum space required for feasible and acceptable timetables? Or: Would forcing staff members to be available for teaching during all lecture hours make timetabling really easier? Answering such questions by trial and error would ask a lot of the timetabling department and consensus among all involved parties on the answers is hard to reach. To make informed choices that are widely supported, they must be supported with some kind of data. Which is the reason to develop a tool to assess the impact of policies on timetables and space usage.

MODEL BEHIND THE TOOL

Our tool is based on a model of Beyrouthy et al. (2006). They found that school buildings' utilisation can be used to predict whether or not a feasible timetable can be found for a given set of events in a given set of rooms. Utilisation of lecture rooms is defined as the fraction of seat-hours that is used, where seat-hours are a multiplication of a number of seats and time. The number of seat-hours needed for an event is the number of attendees multiplied with the duration of the event. The total number of seat-hours is the combined number of seats in all rooms multiplied with the time that they are available for scheduling.

$$Utilisation = \frac{seat\text{-}hours\ used\ for\ events}{total\ seat\text{-}hours}$$

Our tool schedules a set of events under different utilisations to find the utilisation above which no feasible timetable is found. The utilisation changes if either the set of events is altered or the number of rooms is changed. We use the same set of events and try to schedule it multiple times, each time in a smaller set of rooms. Scheduling is automated, in a way that the timetable complies with predefined constraints and events are excluded if they do not fit in the timetable without violating constraints. This can cause a difference between the set of events we are requested to schedule and the set of events that made it into the timetable.

We distinguish two types of utilisation: *Requested utilisation*, which is calculated with the seat-hours of all events in the set and *achieved utilisation*, which only uses the seat-hours needed for events that are included in the timetable. The denominator in the fractions below is in both cases the same but changes with the set of rooms.

$$Requested\ utilisation = \frac{seat\text{-}hours\ needed\ for\ all\ events}{total\ seat\text{-}hours\ available}$$

$$Achieved\ utilisation = \frac{seat\text{-}hours\ of\ scheduled\ events}{total\ seat\text{-}hours\ available}$$

In an ideal situation the achieved utilisation is equal to the requested utilisation until it reaches 100%, which the requested utilisation can exceed unlike the achieved utilisation. However, this is rarely the case, e.g. scheduling four classes in five rooms will always leave one room empty. Other circumstances, like a mismatch between class and room sizes, can also prevent a high achieved utilisation. However the largest contributor to the difference between requested and achieved utilisations are excluded events.

At a certain requested utilisation, the constraints or total room capacity make it impossible to schedule all events. Above that utilisation it is almost always impossible to get a feasible timetable whereas it is almost always possible to do so with a lower requested utilisation. That point is called the critical utilisation and can be identified in a graph. Figure 1 shows an impression of an output graph of the tool. Each dot represents the achieved utilisation after solving the timetabling problem with a particular set of rooms. The critical utilisation is the point at which the graph starts to bend away from the diagonal. In an ideal and fictive situation, all events can be scheduled until the total room capacity is used. The red series in the graph shows results of a test experiment with data from the AUAS. In this realistic scenario, the critical utilisation is much lower and harder to pinpoint. We find in this case that $U_C \approx 55\%$.

WORKING OF THE TOOL

The most important part of the tool is automated timetabling. We looked at timetabling competitions for heuristics that perform well. A Simulated Annealing heuristic by Bai, Blazewicz, Burke, Kendall, and McCollum (2012) finished high on the rankings of multiple competitions. Simulated Annealing is a type of heuristic that is used to find near optimum solutions to discrete optimisation problems. These problems have a large but finite number of solutions, each with its own solution value. The heuristic improves the solution in an iterative process by trying small changes. It either continues from the candidate solution or the current solution depending on a probability that depends on how much worse or better the candidate solution is. The best solution found so far improves over a long number of iterations.

We use a penalty score to determine how good a solution is. A small penalty is put on not using seats in a room during lecture hours while larger penalties are put on constraint violations. Allocating events to a room and timeslot reduces the total penalty score but introducing new constraint violations increases the score much more. Over time as many events as possible will be scheduled by the heuristic while events that cause constraint violations are removed from the timetable. Varying penalty scores per constraint will enable the user to prioritise certain constraints over others. The heuristic will schedule as

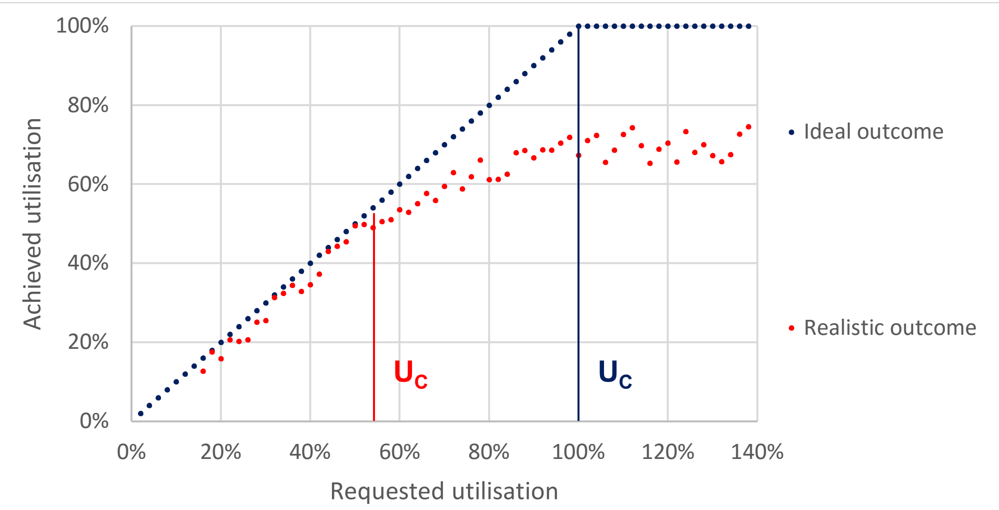


FIGURE 1 UTILISATION PLOT WITH INDICATIONS OF CRITICAL UTILISATIONS

much events as possible while removing events that cause constraint violations from the timetable. Varying penalty scores per constraint will enable the user to prioritise certain constraints over others.

Simulated Annealing needs a starting solution. To create this we gather the information that describes the timetabling problem: What events to schedule, who teaches events, what classes attend them, etc. It is easy to use historic data but it is also possible to create a problem instance from scratch or replace part of the data. Figure 2 shows what input data is needed and for what purpose it is used.

We mentioned that we keep the set of events the same but that we change the set of rooms multiple times. These are created beforehand. As this project is aimed at a specific building of the AUAS, the Fraijlemaborg, the mix of room types and sizes is kept as close as possible to the existing situation. With the set of events, we can calculate how many rooms each set must contain to get an even spread of requested utilisations among the sets of rooms. This spread ensures a clear graph at the end of the process.

Each set of rooms and set of events combination needs its own initial solution. We created them with a simple constructive heuristic: add random events to the timetable in a suitable room during the first timeslot that the room is unused and the classes or lecturers that attend an event have no other events scheduled already. These initial solutions are reused for experiments with different constraints.

After creating the sets of rooms and initial solutions, the user chooses what scenarios are to be compared. By setting the right penalty score per violation for each scenario, the resulting timetables will comply with the right constraints. The output enables the user to compare the achieved and critical utilisations of the scenarios.

INPUT

We tested the tool with data from the academic year 2014-2015. This gave us the opportunity to work with a realistic set of events. The required data is pieced together from multiple sources. The timetable and staff availabilities are exported from their respective software applications and pasted into the model while lecture hours and room characteristics must be looked up and entered by hand.

A variety of constraints has been identified. Most ensure that events are scheduled at suitable locations and staff and students are available to attend. In addition, some scheduling rules have been included that dictate when events can be scheduled. E.g. classes have a minimum number of events per day, or all events are scheduled between certain hours. These rules are based on preferences expressed by students and staff during exploratory interviews.

The penalties per constraint violation are chosen by the user. We created a three tier penalty structure with standard values. This ensures that the model will eliminate the highest tier penalties first, regardless if lower level penalties are induced, and works its way down to the low level penalties. The user is free to choose penalty values but must also ensure that the penalised penalties are compatible. E.g. applying the constraints that a) all events must be scheduled in the morning and b) all events must be scheduled in the afternoon will prevent the heuristic from including any event in the timetable.

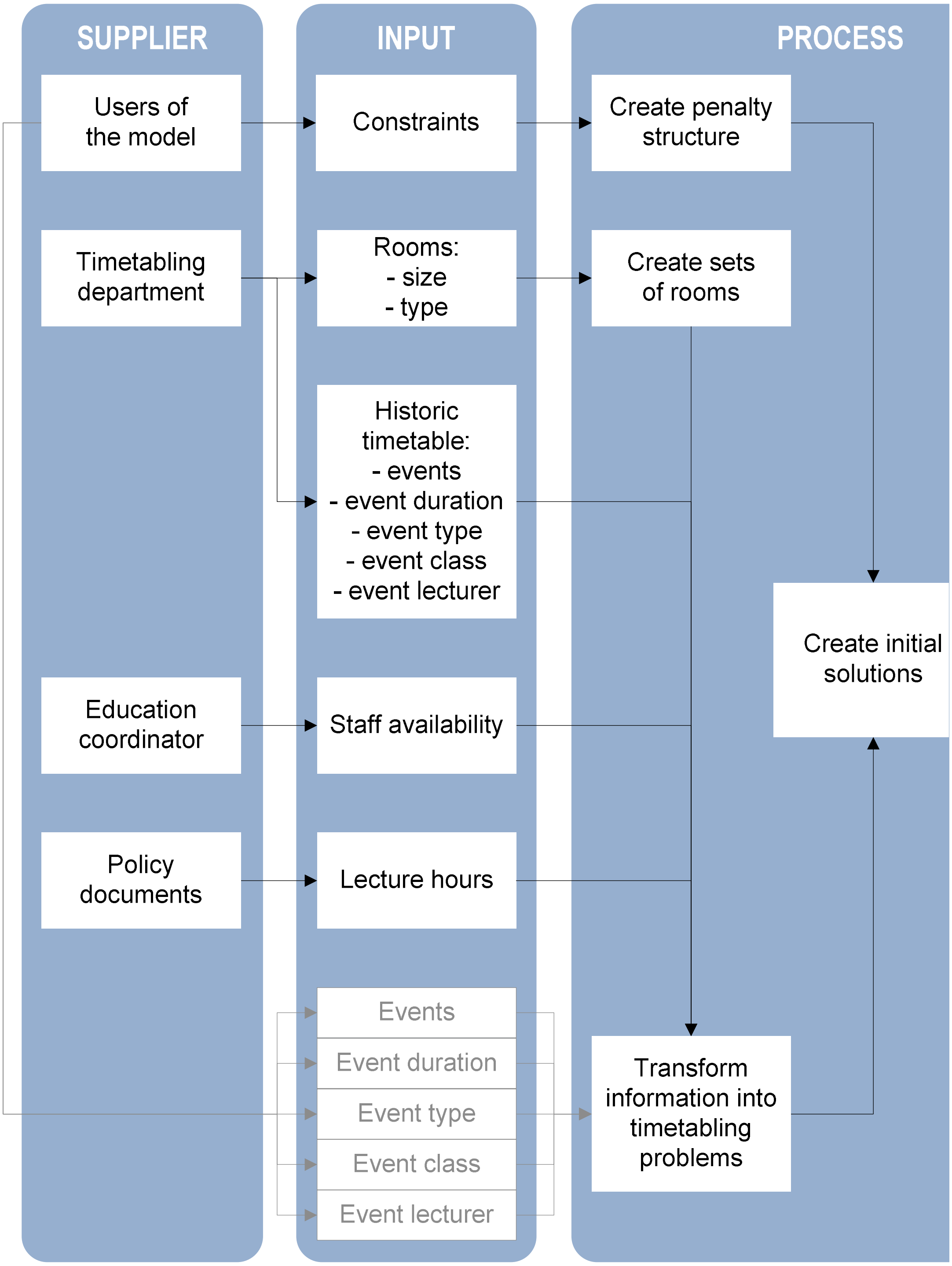
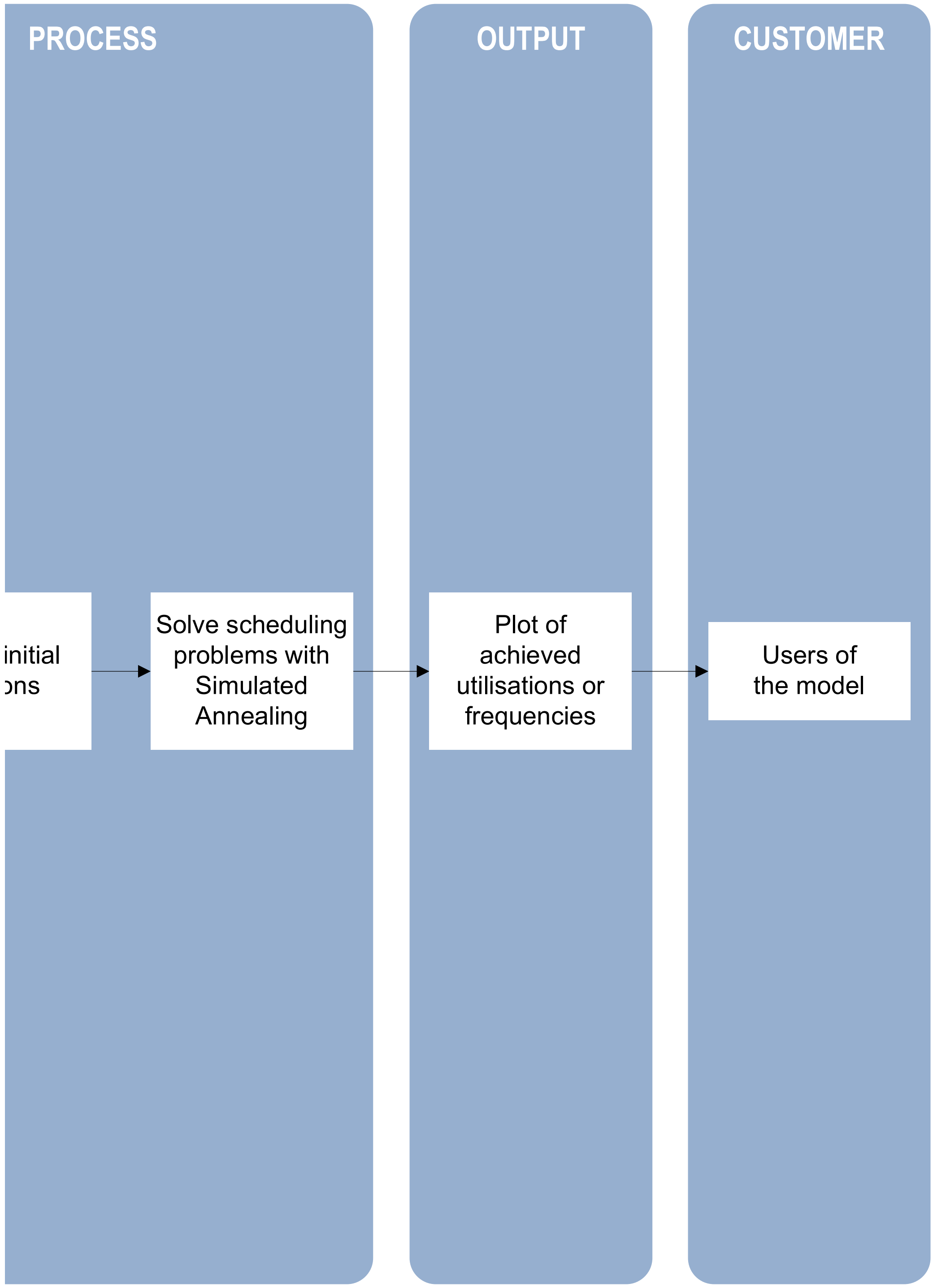


FIGURE 2 OVERVIEW OF THE WORKING, INPUT AND OUTPUT OF THE TOOL


PROCESS
OUTPUT
CUSTOMER
initial
ons
Solve scheduling problems with Simulated Annealing
Plot of achieved utilisations or frequencies
Users of the model

RESULTS

The goal of this project was to develop a tool to assess the impact of policies (timetabling rules) on timetables and space usage. We have created and tested a tool to do so. However, the exact critical utilisation is difficult to determine, as may be noticed in Figure 2. It is important to compare the ability of timetablers through historic timetables with the output of the model, using the same constraints. This will help to match the visually determined critical utilisation to the point that the scheduling department cannot produce feasible timetables anymore.

During the test scenarios that we have run, we found one particular interesting policy that could be implemented: the introduction of morning and afternoon classes. Classes are split in two groups and all events are scheduled in the predetermined day part. This has no negative impact on the critical utilisation while it limits the length of the workday for students, gives students security on when they are expected to attend, far in advance, and provides acceptance for using the first and last few lecture hours on every day.

FURTHER RESEARCH

The outcomes of the tool can be improved by improving the initial solutions to the timetabling process, tailoring them to the constraints that will be applied. Extending the computation time could also result in more accurate results.

An experiment that requires significant preparation but can be done with this model is comparing different curriculum set ups. Balancing resource demand over the academic year across all programmes housed in the Fraijlemaborg could reduce the peak demand for space and staff. Testing multiple curricula requires a new set of events, with staff and class matches and staff availabilities etc. for each instance. This could be an interesting direction to explore.

During the project it became clear that there is little room to increase resource levels. The resources at hand are not fully exploited but improving the actual occupancy of lecture rooms (fraction of used seats in non-empty rooms) could help. This would require research in the difference between number of expected and actual attendees of lectures. Matching the room and class sizes can also help (i.e., splitting multiple large rooms in a larger number of small rooms if there are only small classes).

CONTENT

1 PROBLEM BACKGROUND AND APPROACH

1.1 INTRODUCTION

This research project the completion of the master Industrial Engineering & Management at the University of Twente with a specialisation in Production & Logistic Management. It is carried out at the School of Economics and Management (SEM) of the Amsterdam University of Applied Sciences (AUAS). We will look into the educational logistics of the SEM and try to identify its link to other parts of the educational process.
The AUAS states its mission as follows (Executive Board, 2015):

> *The AUAS is an educational institute that, with a wide range of professional education programmes, trains a large variety of students that develop their talents to the maximum of their ability in order to practice their profession independently and at a high level. Furthermore, the AUAS is a knowledge institution that, through linking education and research, contributes to the renewal of the professional practice and society in and around an internationally orientated Amsterdam.*

It follows that one of the core products of the AUAS is education. Which must be provided to the students within available means and resources. The AUAS also states as one of its ambitions that its students finish their education within the prescribed time (Executive Board, 2015).Which depends on a satisfied, inspired and motivated student body.
The use of resources, personnel and student satisfaction come together in the timetables. The timetables are the result of a trade-off between costs and several resources and can have an impact on the motivation of students to participate in the educational activities on offer. Policies (like no lectures before 11:00 on Monday morning) can increase student motivation and satisfaction but the extra strain on the available resources (empty rooms on Monday, shortages on Tuesday) has been hard to predict or quantify. The goal of this project is to create a tool that enables the school management to do just that: Create a tool that provides insight in the impact of school policy on timetabling and the underlying resources (space and teaching staff).

1.1.1 PROJECT ENVIRONMENT

The AUAS offers 80 bachelor programmes through its seven schools. The school of Economics and Management is divided into four clusters, each with several educational programmes. Two of them have been relocated to one location: the Fraijlemaborg (FMB), these are highlighted in Figure 1.1-1. (Full picture in Appendix 2.)
One of these clusters, Finance & Accounting (FA), initiated this project. It is the largest user of the building and benefits the most from using the Fraijlemaborg as effectively as possible. Collaboration between departments and bodies within the AUAS is becoming one of the policy priorities but the organisation is still compartmentalized and territorial in some areas. The project is done for FA and will be focussed on the programme Business Economics. Outcomes can be produced for other clusters and programmes afterward by using the same methods and tailoring them to specific needs, if necessary.
Cooperation of staff departments is quite important as they are a major source of information. Timetabling is done centralised for all programmes that use the Fraijlemaborg. The scheduling department has a lot of expertise in the field of planning

and has large amounts of historic data on the programmes of FA and the International Business School (IBS).

1.1.2 CORE PROBLEM

The provision of education to students is limited by the availability of time, lecture rooms and teaching staff. Some of these are set centrally by the AUAS like the lecture times while others are decided on and unlikely to be changed like the location of the lectures.
All educational activities of FA and the cluster IBS take place at the Fraijlemaborg and this location will be used for the foreseeable future. Therefore we consider the FMB a given

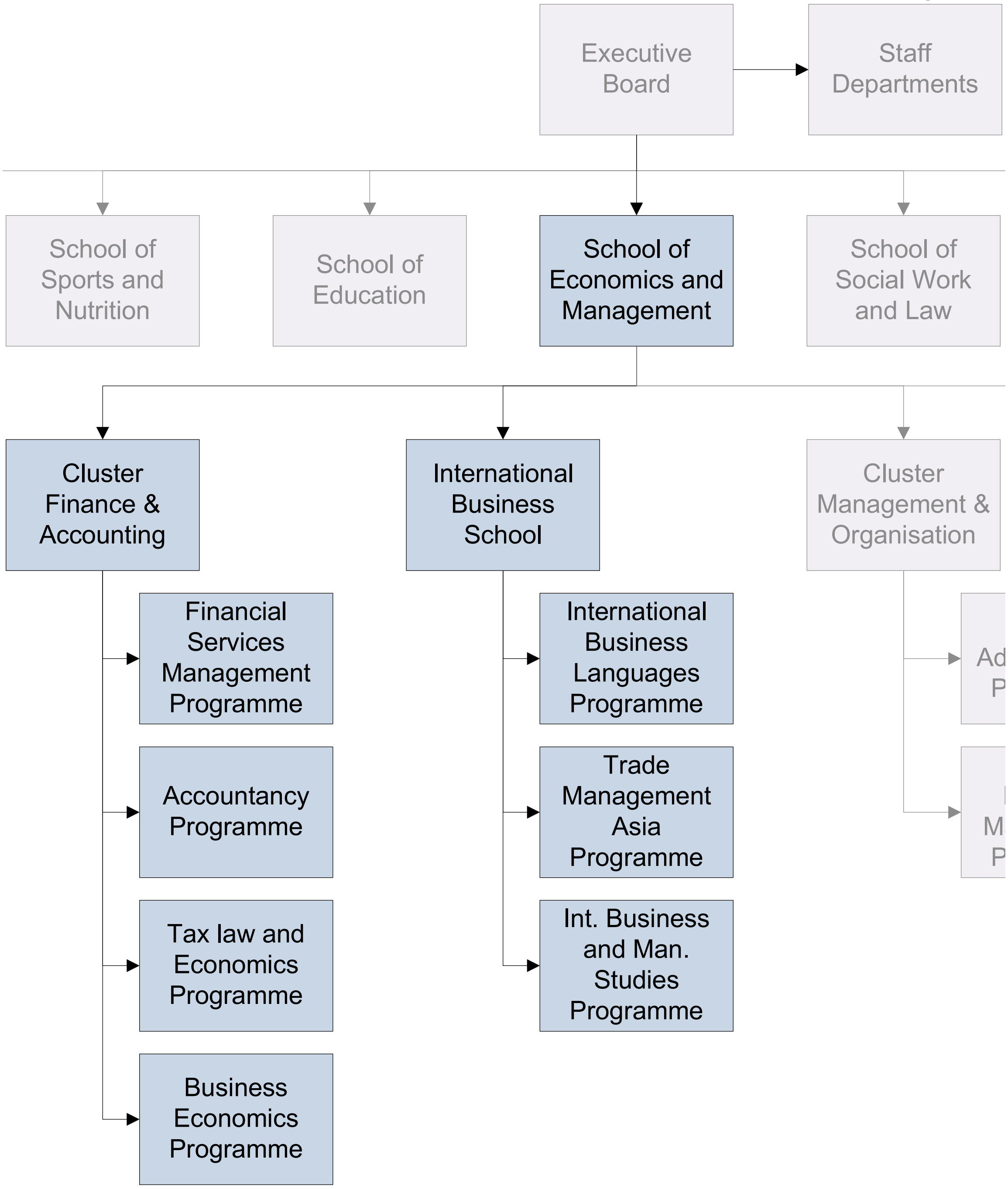


FIGURE 1.1-1 PART OF SEM AT LOCATION THE FRAIJLEMABORG.

for this project. Small modifications to the building can be made but the space is limited to this building and its structure cannot be altered. Another thing that cannot be altered is the content of the programmes or the lecture hours.
Assuming that if students' preferences with regard to their timetable are known, their satisfaction with a timetable and participation in activities can be predicted.
Decisions are made on facilities and human resources, but timetables, where these aspects meet and can be quantified, are not directly included. This is the core problem that is to be solved in the project.

1.1.3 POTENTIAL CAUSES

There may be various issues contributing to the core problem but three of them have been identified as potential main causes. The first of one is the scheduling process. The class schedules are currently made by hand. This process delivers feasible timetables but people throughout the organisation have the feeling that the quality of the schedules can be improved. Such feelings are caused by observations like vacant lecture rooms during highly sought-after time slots. However, assumed underlying problems have not been substantiated yet.
Student dissatisfaction is also noticed through survey results but the reasons behind it are not recorded. To increase the number of satisfied students, one must know what makes them satisfied. To help them finish their degree in time, they have to attend lectures and to do so one has to know what keeps them motivated to attend. As this is not clear, it is impossible to optimise decisions on expected student satisfaction.

The focus within the scheduling department seems to be producing feasible timetables in time. These timetables are currently drafted by hand and automation is put of due to complexity of configuring a software application. This follows from the large amount of timetabling rules and exceptions which must be complete during the input stage to prevent situations that are logical from a technical point of view but obvious errors to the people that use the timetables.
In addition to this situation, problems are caused by the fragmentation of information. Due to the size of the organisation, most information is scattered across lots of departments. This makes it difficult to integrate all constraints and preferences into the timetables. Another complicating factor is the use of multiple copies of the same information by different parties and changes over time, making it hard to assure that the correct version is used.
Besides the availability of information, using the available information correctly is another issue. Creating a timetable is in this case a rather large problem: around 140 lecturers and 80 classes of students must be matched in 98 lecture rooms during a limited number of lecture hours. The lack of a complete scheduling heuristic or algorithm that weighs all constraints and preferences in a structured way makes it difficult to deliver high-quality timetables consistently.

Lectures are scheduled by a central department, detached from the programme and cluster managers. Input information is given after which the timetables are returned as is. This places the scheduling process outside the scope of the daily management tasks.

The potential causes identified so far are:

- Unknown student preferences.
- Complexity of the scheduling process
- Scheduling lies outside management scope.

1.2 RESEARCH APPROACH

This section outlines the objective and research problem of the project, followed by the steps that are taken to solve the research problem and reach the objective. Thereafter follow two paragraphs on data collection and the implementation of the solution upon the completion of the project.

1.2.1 OBJECTIVE

The objective of this project is to create a tool that enables the school management to assess the impact of school policies on the timetables and space usage. The tool will create timetables under different policies and restrictions which can be compared on aspects like student satisfaction and resource utilisation.

In order to do so, we will write a report that maps out the scheduling process and its points of interest. Existing assumptions will be examined and either supported by data or rejected and consensus on what makes a good timetable must be reached with all parties. Furthermore an algorithm will be written. The algorithm is able to construct a timetable based on the constraints and preferences used as input.
The algorithm will be implemented in a piece of software of which the format will be determined later. This will help validate the algorithm and speed up the scheduling process. The increased speed could help in practice but it is intended to enable programme managers to assess potential problems and opportunities in advance. E.g., the effect of a grow or decline in the number of classes, less flexibility due to hiring lecturers that can teach only one subject or concentrating most lectures in the middle of the day can be examined quickly.

1.2.2 QUESTION FORMULATION

The main research question is supported by five sub questions. Answering the sub questions one by one will help to find a solution to the main question that is shown below.

How can timetables and space-use be improved by designing a scheduling tool?

I. *Which of the policies in place influence the timetables?*
II. *What makes a good timetable according to students, teaching staff, scheduling department and school management?*
III. *What are current bottlenecks in the availability of resources?*
IV. *How can user preferences and restricting policies be combined into timetables?*
V. *How can the previous solution be made suitable for the support of decision making?*

1.2.3 ACTIVITIES

The goal is to create a tool that will schedule classes based on input parameters, making the scheduling algorithm the back bone of the tool. Its output will be based on timetables, the actual content of the output (e.g. room utilisation, timetable quality, staffing requirements) will follow from the analysis.

The solution to the problem can be found by going through roughly three phases: analysis of the project environment, design of the management tool and its implementation. Each consists of several activities which are listed below.

Analysis

- Analyse the scheduling process
- Map out school policies that influence timetables
- Stakeholder analysis
- Determine the user preferences
- Review historic data (timetables)
- Literature review on scheduling and timetables

Design

- Compare, select and combine scheduling algorithms and heuristics
- Make a concept algorithm
- Verify the algorithm
- Determine the output of the tool
- Review the available data and choose a fitting software format
- Program the algorithm
- Validate the programmed algorithm

Implementation

- Test the tool with future users (education coordinators of FA)
- Make the report suitable for the support of decision-making

1.2.4 DATA COLLECTION

In order to create the management tool, a wide variety of data is needed. This can be broken down into three parts: resources, policies and goals.

Data on resources is roughly everything that is needed to make a timetable: the number of rooms, lecturers, students, classes and lecture hours per day. But also which lecturer teaches which courses and so on. This data has been used to create timetables and should still be available. The information will be extracted from various systems of the AUAS and supplied by several departments. Difficulties can arise if older data is lost or if departments do not cooperate.

Policies on institutional, school and programme levels will cause restrictions in timetabling and related areas like space management and staffing (e.g. Classes must have a minimum number of contact hours per week or: Teaching space may not be rented at a decentral level). A large number of policy documents has been gathered already and the cluster FA is willing to supply additional information if needed. Other departments may be more reluctant to give information that is not publicly accessible. The second potential problem is that some policies or procedures have been in place for so long that hardly

anybody knows how they came to be. Finding the right person or the right document can be hard in a large organisation like the AUAS, especially as most people are only concerned with the application to their tasks and not the creation of such policies.
The last part is concerns the goals and wishes of people within the organisation: Why do people work at or attend the SEM, what are their needs, what are their wishes, how would they compromise on various things. We will include not only managers but also students, lecturers and other employees that are influenced by timetables. The timetables are the interface between policies on resource use and the educational process. This brings up three major parties: policy makers, users and the scheduling department, see Figure 1.2-1. These three parties approach the timetable from their own perspective and the extraction of information from these parties will differ accordingly.

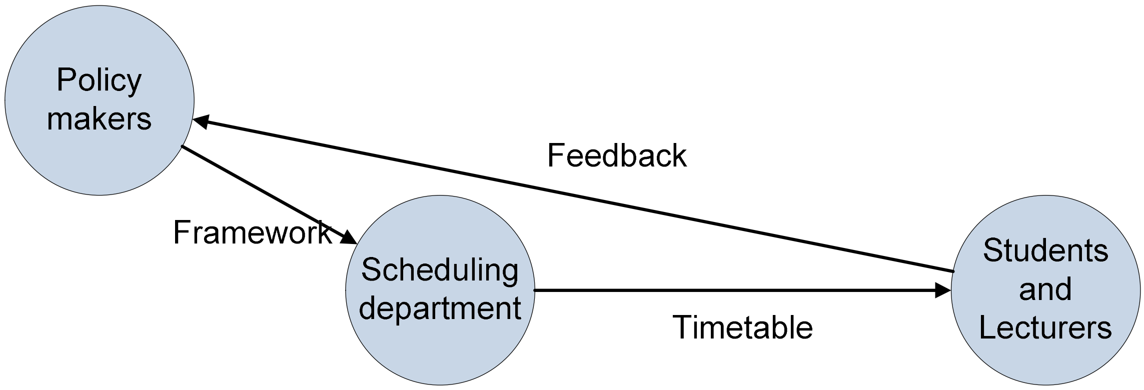


FIGURE 1.2-1 THREE MAJOR PARTIES RELATED TO EDUCATIONAL TIMETABLES.

As each party will have a different relation to the subject, semi-structured interviews will be used to capture all facets of the subject. Surveys can capture a problem but is not suitable to go into detail on possible solutions or the cause.
To ensure that we get the key insights and maybe some surprising additions, we invited opinionated and experienced people: All students are part of the participation council of the BE programme, as is one of the lecturers. One lecturer is member of the central participation council of the AUAS and the other lecturers all have coordinating tasks. The timetabling department is represented by their team coordinator. (We interviewed much more people from the educational branch as their goals can be more subjective and less documented than operational goals.) We try to determine what a good timetable is by interviewing five students, five lecturers, the programme manager of BE and the timetabling coordinator of the SEM. An overview of these interviewees and specialists that are interviewed throughout the project are listed in Appendix 3.
The interviews will be complemented with other sources of information. The general opinions, goals and motivations of the school management are documented in policy documents and scheduling procedures can be found in operating manuals of the timetabling department.

1.2.5 IMPLEMENTATION

The report and tool are primarily intended to support decision and policy making within the school of economics and management. As the group of users is low (education coordinators and manager of FA), the tool can be explained to and tested with the end users. Using the project results unaltered to create timetables by the school's scheduling bureau is of secondary importance and lies outside the scope of this project. However, the scheduling bureau will be involved and the new software will be documented and aligned with existing IT systems as much as possible.

2 MANAGEMENT FRAMEWORK AND STAKEHOLDER ANALYSIS

In this chapter we will sketch a picture of the policies, stakeholders and goals related to timetables and the timetabling process. Section 0 focuses on the management structure of the school and is followed by a stakeholder analysis in section 2.2 and a quick overview of the influence of external factors in section 2.3. The goals and preferences of the stakeholders are presented in section 2.4 The results of the sections above will result in the answers to research sub questions I. and II. in section 2.5 and 2.6 respectively.

2.1 MANAGEMENT AND POLICY

As mentioned before, a lot of things meet in the timetables. In this section we present a management framework to get a quick overview of which planning decisions are made and how these decisions interact.

2.1.1 MANAGEMENT FRAMEWORK

A management framework can be useful to get a picture of all planning processes in the school. Hans, Houdenhoven, and Hulshof (2012) suggested such a framework for the healthcare sector. It shows what planning decisions are made on several hierarchical levels and in different managerial areas throughout the organisation. In their case this was useful to bring clinical staff and administrators together and help them understand which processes are influenced by each other and which planning processes should be improved. Such a division and conflict of interests between the primary and secondary process is also be recognized in the schools of the AUAS (P&O, 2013).
The framework organises all planning decisions in a matrix of hierarchical management levels and managerial areas. To fit the framework to the situation at the AUAS some adjustments must be made. The managerial areas, levels and adjustments are explained below.

Management levels

We discern four management levels: strategic, tactical, offline operational, online operational. Strategic level has the longest planning horizon and concerns the design and dimensioning of resources and processes. There is maximum flexibility and limited data. The tactical level lies between the strategic and operational levels. It is limited by the decisions made on the strategic level but more things are known and certain. The focus lies on the organisation and execution of processes on an aggregated level.
The operational levels cover short-term decision making on the execution of the educational process. These decisions are based on all information that is gathered and constrained by the decisions made on the higher levels. The distinction between online and offline decision making is the online decisions are reactive whereas offline decisions are made beforehand.

Managerial areas

The four areas distinguished by Hans et al. (2012) are medical, resource capacity, materials and financial planning. In our case the first area is easily translated into educational planning and financial planning is also applicable at universities. However we hardly have any consumable resources but two very important and distinct categories renewable resources: facilities and teaching staff. Therefore resource capacity and materials planning are replaced by facilities and personnel planning. The planning choices in these four categories are placed in the matrix in Figure 2.1-1. We derived the planning choices from various policy documents and interviews with staff members. Their area is based on the subject of the decision and their level on the definitions given above.

	EDUCATION PLANNING	FACILITIES PLANNING	PERSONNEL PLANNING	FINANCE PLANNING
STRATEGIC	- Setting the curriculum. - Number of lecture days per week. - Class size. - Lecture time per week.	- Real estate. - Number, capacity and type of rooms.	- Workforce size. - Qualifications and skills of the total workforce.	- Central budgets. - Investment plans.
TACTICAL	- Determining course formats.	- Block planning. - Setting business hours. - Equipment in rooms.	- Balancing staff availability in a week - Assigning lectures to courses and classes. - Training staff members.	- Assign general budgets to projects/ departments. - Decision on what is leading performance or budget and to what extent.
OPERATIONAL (OFFLINE)	- Balancing lectures and course formats over the week.	- Assigning lectures to room and timeslot.	- Scheduling classes and other activities per staff member.	- Cost allocation.
OPERATIONAL (ONLINE)	- Adjusting number of lectures to teaching pace.	- Urgent repairs to the rooms and equipment. - Returning unused rooms to the planning pool.	- Replacing absentees.	- Measuring and adjusting expenses.

FIGURE 2.1-1 MANAGEMENT FRAMEWORK FOR EDUCATION, BASED ON (HANS ET AL., 2012)

2.1.2 MISSING PLANNING FUNCTIONS

Some planning functions may be missing or could be improved. This is often the case but implementing or improving them may yield great benefits.

In order to keep timetables studiable, the workload for students is to be spread across the week. The timetabling coordinator tries to do this but he has the feeling that his information on or perception of the difficulty of courses is incomplete. A dialogue between the teaching staff and timetabling department will help to mix difficult and less demanding lectures better.

Block planning may be useful too. When the curriculum and course formats are known, the demand for lecture rooms per type in hours per week can be assessed. If this is done simultaneously for all programmes of FA and IBS, bottlenecks can be identified and tackled on a tactical level or structurally solved on a strategic level by adjusting the facilities capacity or altering the demand by adjusting and/or aligning the curricula.

2.1.3 CONFLICT OF INTEREST

A key issue is the conflict of interest between the education and operations branches of the school. The first wants to provide students with the best education possible while the latter wants to minimise costs. While teaching staff and educational planning are the responsibility of the programme manager, the facilities and timetabling are controlled by the schools operational manager.

The curriculum is the backbone of the primary process, the educational programme. It dictates which courses are taught in which periods and gives the input for the timetabling process. This process is limited by the availability of rooms that fit the course format and the availability of teaching staff. An ambition of the AUAS is to offer studiable education: a programme that enables motivated students to finish their programme nominally (Executive Board, 2014b, 2014c) and from that perspective the limitations on the timetabling process should be minimized.

There is a theoretical optimum and to achieve corresponding perfect timetables, all lectures of all classes should be given in the best timeslots. Lecture rooms are rented per month and not by the hour; therefore the minimum rented space is equal to the peak in use. This peak can be lowered by spreading the lectures evenly over all available timeslots. Figure 2.2-1 illustrates a simplified trade-off between perfect studiability and minimization of rented floor space.

The spread of lectures can be limited by staff availability and preferred working hours. The operation bureau could improve the studiability of the timetables or reduce the housing costs if the availability of the staff is maximized. In the quality standards for timetabling, the Executive Board (2014c) poses that teaching staff preferences are to be made ancillary to studiability of the timetables.

Changes in facility capacities are directly related to the availability of personnel and decisions on these changes should be made in conjunction. However it is important to first balance the curriculum such that the number of lectures and lecture formats is as constant over time as possible and to determine a target for studiability. Such a target will prevent capacity reductions, of facilities and personnel, to the point that the primary process becomes compromised.

2.2 STAKEHOLDER ANALYSIS

There are a lot of parties involved in the educational process, almost of all of which deal with timetabling. By mapping them out we get to know which forces are at play and how these stakeholders should be taken into account.

2.2.1 STAKEHOLDER ANALYSIS IN LITERATURE

The stakeholder theory was popularized by Freeman (1984). A stakeholder perspective boils down to the idea that a corporation should consider all stakeholders related to the company instead of just considering, suppliers, customers, owners and employees. In this context a stakeholder is defined as:

"any group or individual who van affect or is affected by the achievement of the organization's objective" (Freeman, 1984)

Three decades later, a lot has been written on the stakeholder theory, it has been expanded, and refined by numerous contributors. The topic has grown tremendously but they also led to numerous definitions and views and there is even no consensus on what a stakeholder is (Miles, 2012). Clear evidence that using the stakeholder theory increases an organisations performance also lacks (Donaldson & Preston, 1995). All in all, basing a firm's policy entirely on stakeholder theory is not without controversy. However, this does not mean that it cannot be very useful.

12 Number of courses taught at a timeslot.

All lectures in best timeslots according to students

Lecture hour	Monday	Tuesday	Wednesday	Thursday	Friday
1					
2					
3					
4	20	20		20	20
5	20	20		20	20
6	20	20		20	20
7					
8					

Required rooms: 20
Timetable quality: High

Minimal number of rooms required

Lecture hour	Monday	Tuesday	Wednesday	Thursday	Friday
1	6	6	6	6	6
2	6	6	6	6	6
3	6	6	6	6	6
4	6	6	6	6	6
5	6	6	6	6	6
6	6	6	6	6	6
7	6	6	6	6	6
8	6	6	6	6	6

Required rooms: 6
Timetable quality: Poor

Compromise between lecture times and space.

Lecture hour	Monday	Tuesday	Wednesday	Thursday	Friday
1					
2	4	5	4	5	4
3	6	10	10	10	9
4	12	12	12	12	12
5	12	12	5	12	9
6	10	10	5	10	3
7	6	6	4	6	3
8					

Required rooms: 12
Timetable quality: Average

FIGURE 2.2-1 ILLUSTRATION OF SPACE REQUIREMENTS PER LECTURE HOUR FOR TIMETABLES OF DIFFERENT LEVELS OF QUALITY.

In this project we are not running for profit firm but a budget driven organisation with good education as its product. At first glance, its objective and all powers at work seem vague. By using a stakeholder management framework, all stakeholders can be identified and classified in order to manage the organisation efficiently and effectively. Such a framework is proposed by Mitchell, Agle, and Wood (1997) who based their work on the wide variety of views on stakeholder theory and management available at the time.
In this framework each stakeholder has some stake or claim on an organisation: an interest that they want to have served. Managers within the organisation can address these issues. However, due to time and resource constraints, not all claims can be attended to. Therefore matters that are important, salient, to managers will be addressed first. The salience of (groups of) stakeholders can be determined by categorizing them on the basis of three

attributes. The definitions of these three attributes: power, legitimacy and urgency can be found in Table 2.2-1.
Figure 2.2-2 shows the stakeholder types that follow from the possible combinations of attributes. These types can be divided into three categories: latent, expectant and definitive stakeholders, all of which will be explained below.

TABLE 2.2-1 KEY CONCEPTS IN STAKEHOLDER IDENTIFICATION BY MITCHELL ET AL. (1997)

Construct	Definition
Stakeholder	Any group or individual who van affect or is affected by the achievement of the organisation's objectives.
Power	A relationship among social actors in which one social actor, A, can get another actor, B, to do something that B would not have done otherwise. *Bases* *Coercive* (Force/Threat) *Utilitarian* (Material/Incentives) *Normative* (Symbolic/Prestige/Esteem)
Legitimacy	A generalised perception or assumption that the actions of an entity are desirable, proper, or appropriate within some socially constructed system of norms, values and beliefs.
Urgency	The degree to which stakeholder claims call for immediate attention. *Bases* *Criticality* (The importance of the claim or the relationship with the stakeholder.) *Time sensitivity* (The degree to which managerial delay in attending to the claim or relationship is unacceptable to the stakeholder.)
Salience	The degree to which managers give priority to competing stakeholder claims.

Latent stakeholders

The outer ring in Figure 2.2-2 consists of latent stakeholders. These only have one of the attributes and are therefore deemed unimportant or are even disregarded by managers. The stakeholders in categories 1, 2 and 3 are often passive.
Dormant stakeholders have power but no legitimate or urgent claim are not important yet. However managers should be aware of them as they can acquire urgency or legitimacy over time.
Discretionary stakeholders lack power and urgency but as their claim is legitimate, addressing their claims is positive. Doing so is based on goodwill as managers will not experience pressure to do so due to the lack of power or urgency.
Demanding stakeholders are a nuisance to management as they are vocal about their claim. They can however be ignored as long as they have no legitimate claim and no power to act themselves. An example would be protesters for a cause without broad support.

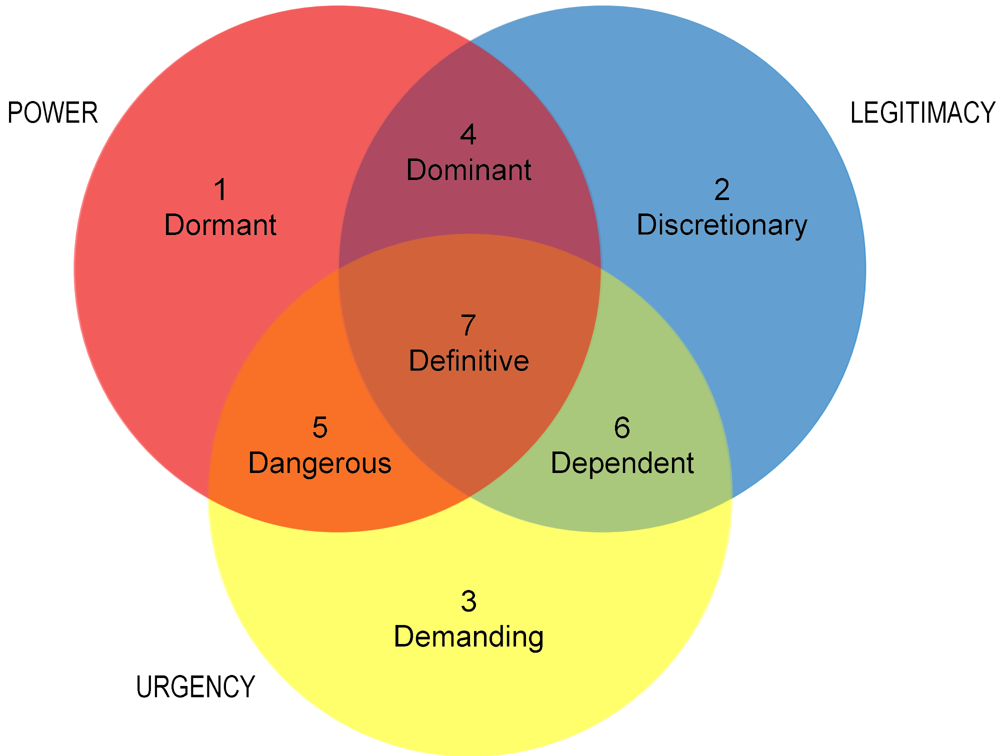


FIGURE 2.2-2 STAKEHOLDER TYPES AND THE THREE MAIN CHARACTERISTICS (MITCHELL ET AL., 1997)

Expectant Stakeholders

The stakeholders in categories closer to the centre, 4, 5 and 6, combine two attributes. This makes them more active and moderately salient to managers. They are recognized and dealt with by management.

Dominant stakeholders pair legitimacy with power. Their legitimate claims should be addressed and their power makes sure that managers must buckle in the long term. Dominant parties can therefore easily attract management attention.

Dangerous stakeholders may have illegitimate claims but are willing to exercise their power to fulfil their claims. Managers need to identify them to mitigate any danger to their organisation or other individuals and entities involved. An example of dangerous stakeholders is a group of employees that sabotages their production to extort exorbitant wages.

Dependent stakeholders have urgent and legitimate claims which ought to be addressed by management. However due to the lack of power, these claims will only become highly salient if they are backed by managers or other stakeholders with power.

Definitive

Definitive stakeholders have power, legitimacy and urgency and will have a high salience. Managers must address their claims immediately. Any expectant stakeholder can acquire their missing attribute to become definitive. Most of them however will be dominant stakeholders whose claims become urgent. E.g. departmental budget overruns start to affect the central budget significantly, making the interests of the central directors urgent.

The three categories mentioned above help managers to find which stakeholders should be served first. However, the attributes of a stakeholder can change over time and their salience will change with it. Some of these changes can be predictable. Furthermore not all stakeholders are conscious of the attributes they have (e.g. students are unaware of the financial effect of dropping out if they are dissatisfied with their education). If the stakeholders and their attributes are mapped out and updated, managers can include stakeholders in their decision processes they may overlook otherwise or will become highly salient in the near future. We will make such an overview of the school in the next section.

2.2.2 STAKEHOLDERS OF THE TIMETABLING PROCESS

We start off identifying potential stakeholders by mapping the timetabling process, see Figure 2.2-3. However not all stakeholders are covered (e.g. students who do have a claim on the timetables but are not involved in making them). In Table 2.2-2 an overview of all stakeholders. These include all parties with one or more of the attributes mentioned in section 2.2.1.

Definitive stakeholders

We identified three definitive stakeholders: the programme and operations manager and the students. The first two represent the two managerial areas with a conflict of interest that were mentioned in the section 2.1.3 while the students are not directly included anywhere in the process.
The key to solving any problems related to timetabling is giving both managers a joint goal and explicitly include the interest of students.

Expectant stakeholders

The majority of the expectant stakeholders are Dependent stakeholders: the timetabling, education and team coordinators. Their lack of power slows the gathering of input information and therefore the start of the timetabling process. Backing of a stakeholder with power (the dean or one of the managers) could help to get people to act on requests of the coordinators. Especially the backing of the timetabling coordinator by the programme coordinator could change his salience and the whole timetabling process dramatically.
The last expectant stakeholder is the dean of the SEM. If problems surrounding timetabling become critical to the performance of the whole school or if a dependent stakeholder gets his attention, the dean will instantly become a definitive stakeholder. The involvement of the dean in matters at an operational and tactical level can be monitored to predict if or when he shifts the power balance.

Latent stakeholders

There are two latent stakeholders left, both dormant: the schedulers and the lecturers. The teaching staff has an open dialogue amongst each other and with the team coordinators so if matters become urgent, they will probably be recognized in an early stage. The timetable schedulers on the other hand work quite isolated and could acquire urgency or legitimacy undetected. Here lies also a chance for both the operational and educational branch: if the schedulers can be inspired to actively pursue either legitimate goal (high quality education or minimization of costs) they become dominant stakeholders and become more salient.

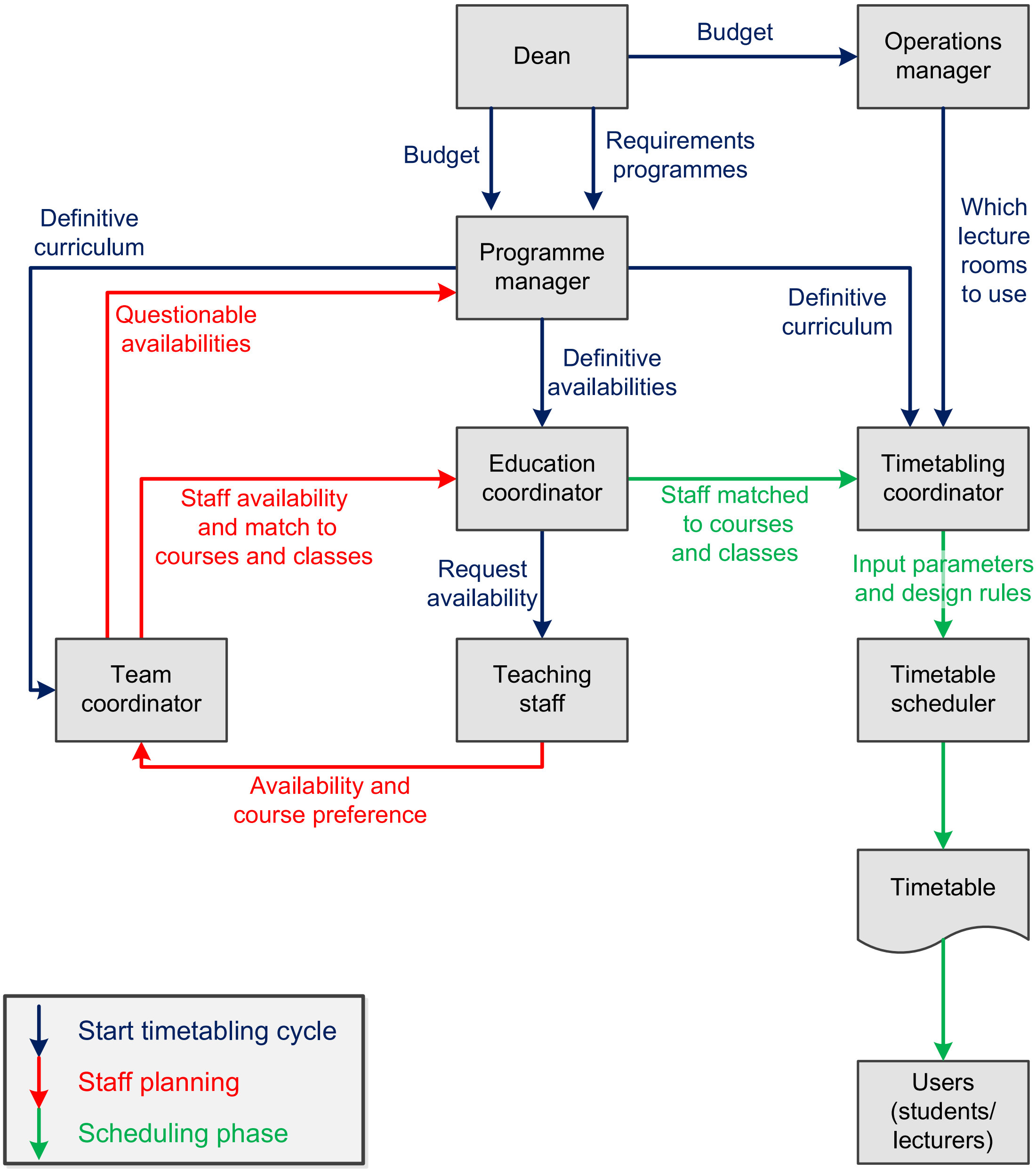


FIGURE 2.2-3 THE ACTORS IN THE TIMETABLING PROCESS AND INFORMATION FLOWS, BASED ON INTERVIEWS.

TABLE 2.2-2 OVERVIEW OF THE STAKEHOLDERS OF THE TIMETABLING PROCESS

Stakeholder	Stakeh. type	Attributes	
Dean	Dominant	*Claim*	Balance the finances between education an operations.
		Power	Coercive: Superior of all employees of the school. Utilitarian: Decides on budget allocation
		Legitimacy	Creates the environment for the programmes to flourish.
		Urgency	No
Operations manager	Definitive	*Claim*	Minimizing operational costs.
		Power	Coercive: Superior of all members of the operations bureau. Utilitarian: Controls the lecture room capacity.
		Legitimacy	Yes, facilitates the primary process of the school.
		Urgency	Low housing costs are critical to manage operations within budget.
Programme manager	Definitive	*Claim*	Offer a high quality educational programme to the students.
		Power	Coercive: Superior of teaching staff, team- and education coordinators. Utilitarian: manages budgets and staff deployment (related to the programme).
		Legitimacy	Yes, manages and represents the primary process.
		Urgency	High quality education is critical; it is the core product of the institution.
Timetabling coordinator	Dependent	*Claim*	Gathering input parameters and establish timetable design rules
		Power	No
		Legitimacy	His work helps to create good quality timetables.
		Urgency input	Timetabling and therefore gathering its is time sensitive.
Timetable schedulers	Dormant	*Claim*	Finding a schedule based on the constraints and requirements known at the moment of scheduling.
		Power	Utilitarian: controls the output of the timetabling process.
		Legitimacy	No, no claim to the underlying primary or secondary processes. They try to find a feasible solution within the framework of constraints they are presented with.
		Urgency	No

Stakeholder	Stakeh. type	Attributes	
Education coordinator	Dependent	*Claim*	Gathering staff availability and information on course content.
		Power	No
		Legitimacy	Work contributes to good and workable timetables
		Urgency	Timetables cannot be made without the gathered information.
Team coordinator	Dependent	*Claim*	Balancing staff availability and assigning staff to classes and courses.
		Power	No
		Legitimacy	Good staffing helps improving the education and timetable.
		Urgency	Getting the staffing right is critical to the quality of the primary process and the quality of the timetables.
Teaching staff	Dormant	*Claim*	Pleasant working hours, lecturing preferred courses and a reasonable workload.
		Power	Utilitarian: Resignation reduces teaching capacity of the school.
		Legitimacy	No, just individual preferences.
		Urgency	No
Students	Definitive	*Claim*	Studiable and pleasant timetable.
		Power	Utilitarian: Can reduce revenues by dropping out. Normative: Determines whether targets are met by the school through the NSE.
		Legitimacy	Expecting a studiable timetable that helps them perform well is desirable.
		Urgency	Their performance and opinions are critical to the institutional goal: providing society with highly educated professionals.

2.3 EXTERNAL ENVIRONMENT

We have limited ourselves to the school while identifying stakeholders, everything outside the school is in that sense the external environment. This environment will influence certain decisions, mainly by the dean and operations managers. Their budgets are dictated by the government and their autonomy with regard to housing is limited by policies and availability at a central level of the AUAS.

2.3.1 ROLE OF FACILITY SERVICES

Because of its large influence on housing choices, we interviewed two people from Facility Services: one project worker Educational Logistics and a staff member Real Estate

Management. We covered space availability, costs and the determination of space demand.
Facility Services manages the real estate and housing needs of the AUAS and provides accomodation at various locations in Amsterdam. For example on the newly built Amstel Campus or in case of FA at the Fraijlemaborg. Some accomodations are owned by the AUAS while others are rented (with longterm contracts). They try to fit all departments in accomodation of suitable size and facilities such that the available buildings are used fully and rental accomodations are kept to a minimum. All schools come to an agreement with Facility Services on which accomodation to use. Schools are not free to pass on an offer and rent space from external parties.
The housing costs are based on a AUAS wide price per square meter. Following the interview with staff from Facility Services, Jeroen Roosen looked into the formation of that central price. It was found that the costs of certain central services are divided between the departments with Activity-Based Costing since 2012. Costs of the services are allocated to activities and these activities are given units of account. E.g. rent and depreciation are related to space use and space use is measured with square meters of functional floor space. The costs in the example and others that have floor space as unit of account are all added up and charged to the schools and departments pro rata with their space usage.
These AUAS-wide prices are based on solidarity. Costs (and prices) are optimised on university level. Which drove the costs up for departments, programmes and faculties that had below average costs before 2012 as local cost optimisation that increase the total costs is prohibited.
Space demand is determined with a space standard, a certain area per student. This standard was determined in 2008 after research by Van Aarle De Laat. The original documentation was not available anymore but Jeroen Roosen did find out that it was based on the space use and student numbers in 2008: It was estimated that, with some efficiency gains 1.3m^2 per student would be enough for all regular lecture rooms (0.9m^2) and office space (0.4m^2). This original standard is also used for budgetting purposes. Faculties are free to deviate from the standard as they see fit. (E.g. if a programme offers a below average number of contact hours to its students.)

2.3.2 IMPLICATIONS OF THE SPACE STANDARD

As mentioned briefly: the vast majority of funds are provided by or set (tuition fees) by the government and split between the faculties. Making the AUAS as well as the SEM a budget driven organisation. The funds available dictate what resources can be used instead of basing resource levels solely on demand. This combined with a rule of thumb/standard of unclear origin can provoke the thought that the operational branch does not determine the space capacity based on what is needed to provide a good educational programme.
Detailed and complete numbers on space use and costs per programme over a longer period of time proved hard to come by. We managed to piece together an overview of the development of student numbers, staff levels and space usage of SEM between 2010 and 2016 (Figure 2.3-1). The underlying number come from the AUAS budgets between 2012 and 2015 and a few management reports, the numbers for 2015 and 2016 are projections for all figures.

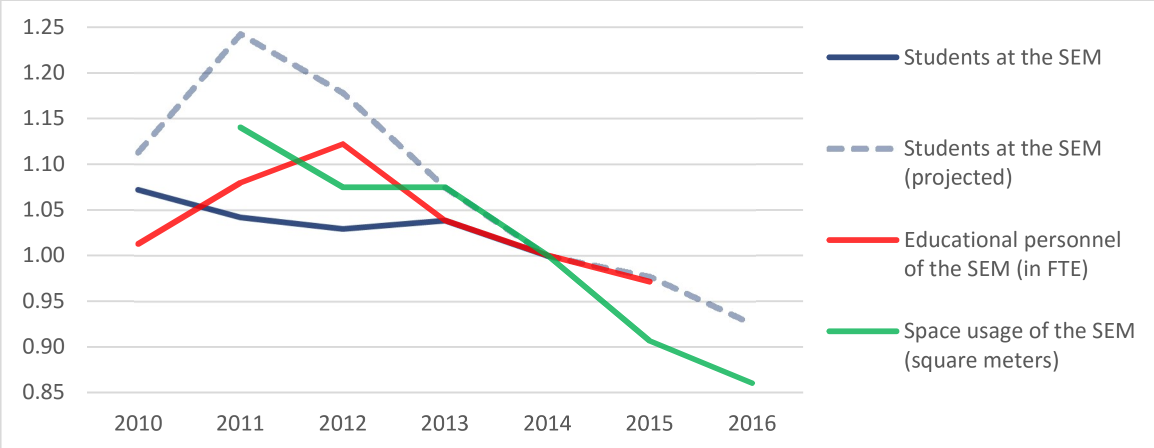


FIGURE 2.3-1 RELATIVE CHANGE IN STUDENT NUMBERS, STAFF LEVEL AND SPACE USAGE (THE LEVELS OF 2014 ARE SET TO 1, BASED ON (EXECUTIVE BOARD, 2011, 2012A, 2013A, 2014A), AND MANAGEMENT REPORTS).

The staff levels seem to follow the projected student numbers albeit with some delay and the student/lecturer ratio seems stable since 2013. The other main resource level, space, seems to decrease twice as fast as the number of students. It dropped from 1.421 m^2 per student in 2011 to 1.298 m^2 per student in 2014 and is projected to decrease even further: below the standard. This implicates that the space standard is not the bare minimum which leaves the question: What is? While that question remains unanswered, the resource levels and quality of education can creep slowly to unacceptable levels. The result of this project is a tool that can quantify what space usage is needed to accommodate a predetermined acceptable educational process. This should help to identify hidden, structural problems. Problems which then can be solved or dealt with in a trustful and transparent environment.

2.4 STUDENT AND STAFF PREFERENCES

Through several interviews the preferences of students and staff members have been determined. An overview of who were interviewed can be found in Appendix 3. The outcomes are presented below per group.

2.4.1 PROGRAMME MANAGER AND TEAM COORDINATORS

At the centre of the educational branch are the programme manager and team coordinators. Their primary goal is to provide students the best education possible. The starting point is the curriculum which is built up in three lines: knowledge, skills and formation. Courses are sequential in those lines and those lines cover the entire length of the programme. Most courses include multiple formats (lectures, computer tutorials, project guidance etc.) which require different resources (types of rooms). The demand for resources is and will not be perfectly balanced over the year as this would deteriorate the quality of the programme.

Another important goal is to encourage interaction between students and lecturers. This is facilitated by matching classes and lecturers for a longer period of time. Students will be taught by the same lecturers in several periods. E.g. their second year statistics module is ideally taught by the same lecturer as their first year statistics module. This makes the teaching staff more approachable and more closely involved.

This leaves student participation in the courses. An average student ought to spend 42 hours per week on his education to finish his or her programme nominally. To encourage this their lectures (around 16 hours) are to be spread over four days to keep them engaged with their studies and give them one day a week to work on their coursework and assignments. However spreading the work to keep student focused on their studies is not the only factor, their attentiveness during lectures is also important. Therefore, for example, the first and last lecture hours should be avoided.

In addition to these education related matters, there is one more general wish. There is a feeling that the programmes are not equally assigned highly sought after and unwanted timeslots and lecture rooms.

All in all, the curriculum is set and there is no intention to alter it to spread resource demand evenly and there are some preferences regarding the timetable. These are spreading the lectures over 4 days per week for every class, avoiding early and late lectures and dividing the positively and negatively rated timeslots and rooms equally among the programmes that use them.

2.4.2 STUDENTS

The students seem to want a day programme that is worthwhile to attend. A major issue for students is travelling time. One way commutes between 40 and 90 minutes are normal. This makes it very tempting to skip a 100 minute lecture that could seemingly be read at home too. Therefore four lecture hours is mentioned as the minimum number of lectures hours per day and lectures a preferred between 10:00 and 16:00

In order to be home at a decent time or to have the freedom to choose to work on assignments, it is important to concentrate the lectures close together. However, free periods are not necessarily a problem. Students use them to work on assignments or projects. In this case consecutive free periods are preferred: rather two and four hour lectures with a two hour period in between and three two hour lectures separated by two single hour free periods.

Once more, it is important to make it worthwhile to attend a lecture. Courses that are difficult or related to the field of study are deemed more important and are less likely to be skipped. These can be scheduled on their own, while courses that are deemed less interesting or important are better scheduled adjoined to others.

Probably due to the low number of scheduled hours a week, combinations of hard classes were not widely mentioned as a problem. A tough day was more often described as a long (eight hour) day with hardly any breaks.

In summary, the students' preferences with regard to their timetable are: lecture times between 10:00 and 16:00, no multiple single free hours on a day, lectures clustered and no long days without breaks. In order to keep them from skipping class, courses that are perceived to be less important should not be scheduled apart from other lectures.

2.4.3 TEACHING STAFF

Most lecturers have some personal preferences with regard to their courses and working times. However they stress that it are preferences and not demands and should not be leading in scheduling.

The things they find important are related to the students. Lecturers prefer times that do not negatively affect attendance. Furthermore they would like to have at least a day between lectures of the same course in order to let the students process the material.

2.4.4 TIMETABLING TEAM

The timetabling team is focused on scheduling all classes in accordance to the given constraints. Where the constraints are: the lecture rooms to be used, the staff availability and the lecturer-class matches. While doing so they try to accommodate wishes like avoiding the first and last lecture hours.

From an operational perspective, the maximum number of rooms/floor space should be minimized to reduce housing costs. This is a constant pressure. To do so and still find feasible solutions to the timetabling problem, staff availability should be maximized while the number of other staff related constraints is to be minimized. This wish is the core of the conflict by education and operations. The educational branch feels that it has reasonable demands and that the support staff, the schedulers, should help them by providing the best timetable that fits within their constraints whereas the schedulers feel that their job is made needlessly difficult by most parties involved.

The team links information together into a timetable. In order to do so they require complete and correct information on time to start scheduling. The focus is on necessary information. However they think they can refine timetables if they get inessential information like which courses and course formats are to be scheduled far apart and which can be scheduled consecutively without problems.

2.5 SUMMARY OF POLICIES

The sections below will answer research sub question I. per managerial area of the framework in section 2.1.1. This question is:

I. Which of the policies in place influence the timetables?

Most policies mentioned below are formally documented, however, some exist as tacit knowledge in the organisation.

2.5.1 EDUCATION

There are few formal policy documents on how the educational program is to be organised. Ideas and didactic concepts are discussed in teams and incorporated in curricula, courses etc.

Curriculum

Curricula are set for multiple years and contain the outline of the programme: What is taught, at which stage of the programme. They can be revised but programme managers tend to avoid that as it will requires a lot from the educational organisation besides the regular teaching activities.

Performance Agreements with the Government

The AUAS makes performance agreements with the Ministry of Education, Culture and Science. Part of these agreement is a minimal number of contact hours between lecturer and student (Executive Board, 2012b, 2013b). The minimum lecture time per week that a programme should offer is dictated by these agreements.

Informal policy

Programme manager, team coordinators and course coordinators work from the existing situation and alter it directly if deemed necessary, without the need for formal documentation. These decisions affect all managerial levels from the number and size of classes, via course formats to balancing the workload of students over the week during timetabling.

2.5.2 FACILITIES

Lecture rooms are the main facilities needed for timetabling. The area of teaching spaced is to be based on AUAS wide guidelines (Facility Services, 2013). However the operations manager of the school may deviate from these guidelines. The bridge between operations and education in the form of a service level agreement (as advised by Van Beek (2012)) is not yet in place.

2.5.3 PERSONNEL

Personnel planning with regard to teaching staff is the responsibility of programme managers. They have freedom on the composition of the workforce but are restricted in its size.

Budget of the AUAS

The AUAS reserves a certain amount of money to pay teaching staff of each school (Executive Board, 2014a). This budget limits the size of the workforce.

Performance Agreements with the Government

The Executive Board (2012b) has agreed that a certain percentage of the teaching staff has a master's degree. This limits the candidates a programme manager can hire.

Collective Labour agreement

Collective labour agreements limits the number of working hours that can be assigned to employees. These set the boundaries, in combination with task assignments, to the deployment of staff members.

Task assignment Standard

The standard assigns a time load to all tasks of a lecturer. This helps to objectively spread the workload over the staff and influences the task assignment that influences the deployment of staff members.

Informal policy

Assigning courses and classes to lecturers (who does what) are done by team coordinators in conjunction with the lecturers involved.

2.6 SUMMARY OF PREFERENCES

This section answers the second research sub question:

II. *What makes a good timetable according to students, teaching staff, scheduling department and school management?*

In separate interviews, the students came in broad lines to the same ideas. Staff members differed more in the solutions they proposed but all aimed at the same goal: high attendance and attentiveness.

Students want to use their time efficiently. They weight travel time with the amount of lectures and their necessity to decide whether they come to the Fraijlemaborg at all or not.

Once there they want to make use of their time so they do not want to cut off project meetings because the next lecture starts and have another free period afterwards. The decision on attending a lecture is also often made for a separately planned first or last lecture of the day.

Staff members concentrate on getting students to work on their studies for 40 hours a week (time on task), the absorption and processing of the learning material. I.e. getting students to attend and make use of the lectures offered to them.

Table 2.6-1 contains the wishes and preferences of the schools staff and the students. It will be the basis of the scoring of timetables in the following chapters. The last two rules were not directly mentioned but can help to solve conflicts between students and staff: Scheduling three days of lectures per week of which Monday and Friday are given, makes each of them worthwhile to travel for while avoiding the loss of an entire day of studying as students get the feeling that it is weekend already.

TABLE 2.6-1 STUDENT AND STAFF TIMETABLING PREFERENCES

Stakeholder	Wish or preference	Motivation
Operational manager	A flat demand for each resource over the year	Minimum housing costs
Programme manager	All classes (of all programmes) have equally good timetables	Offering quality to everyone
Staff	Mixed course formats per day	Attentive students
Staff	Lectures spread over four days per week for each class	Increased time on task
Staff	No lectures of the same course on consecutive days	Allow students to process the matter
Staff	Lectures in hours 3-10 (between 10:20 and 16:10)	Attentive students and high attendance
Students	Lectures in hours 3-10 (between 10:20 and 16:10)	Fit in with private life
Students	In case of multiple free periods on a day, join them together	Allows to focus on group or course work
Students	At least four lecture hours on a day	Makes the commute worthwhile, higher attendance
Students	Total lecture day spans no more than 8 lecture hours.	Attentiveness
Students	Mixed course formats per day	Attentiveness
Students	Timetable the same per week over a long period of time	Fit in with private life (job)
Students	Timetables available three to four weeks in advance	Fit in with private life (job)
	Schedule easy/unimportant courses subsequently to other courses	Higher attendance
	Lectures on the Monday and Friday for each class	Increased time on task without the four day constraint

3 UNDERLYING MODEL

This chapter explains on which model the tool is based in section 3.1 and the working of the timetabling heuristic that we used in sections 3.2 and 3.3. We show what input data we used for the tool in the final section, 3.4,.

3.1 DETERMINING THE USAGE OF EDUCATIONAL SPACE

The objective of this project is to create a tool that enables the school management to assess the impact of school policies on the timetables. Policies determine demand and resource capacity. E.g. policies like “all lectures must be taught between 12:00 and 14:00”, “only fifteen lecture rooms may be used” or “at least 50 accountancy lecturers must be employed” create an extreme peak demand for lecture rooms, a very low space capacity and a high teaching capacity respectively. The demand for space and staff follows from the educational events (all lectures, tutorials, etc. in the curriculum) that are to be scheduled. While the resources teaching staff and lecture rooms are determined by things like the number of employees and their days of and the amount of floor space rented.
In order to see how demand and resource levels interact, we vary them and evaluate which combinations make it impossible to create a feasible timetable. In timetabling the main resources are space and teaching staff. It is often accepted that space remains unused sometimes while staff is almost always utilised fully. This makes educational space the closing entry:How much space is needed given the classes, curricula, business hours and staff levels?
In order to determine the required space, we must determine when the available space is used to its full potential.

3.1.1 DEFINITION OF SPACE UTILISATION

Lecture rooms (educational space) offer a place to teach a course to a group of students. They basically consist of a quiet environment with seating arrangements and some means of communicating the course content. But the usage of a school building can be measured in various ways, e.g. is it fully used if all rooms are used all the time of only if all seats in all rooms are used each lecture hour? Furthermore, on an even lower level what is the unit of measure of the capacity of a lecture rooms, its floor area, its volume or the number of seats in it.
To start with the latter, Beyrouthy et al. (2006) use a unit of measure that can be used to measure both the capacity of a room and the demand generated by an event: the seat-hour. It is a multiplication between the number of seats or students with a period of time. E.g. a room with 30 seats in a school that operates 5 days a week, 8 lecture hours per day has a capacity of 1200 seat-hours (per week) while a two hour lecture for a class of 25 students creates a demand of 52 seat-hours. They also use roomslots as a unit of measure: The number of timeslots a room is available or used. E.g. if we have again, 5 days a week, 8 timeslots per day and have scheduled a two hour lecture, then there are 80 roomslots in total of which 2 are used.
In timetabling, events are scheduled in rooms during a period of time. Such a period must often fit set lecture hours. We call each of these possibilities (lecture hours) a timeslot.

Beyrouthy et al. (2006) also identified three important measures of space usage on an aggregate level: *utilisation*, *frequency* and *occupation*. They are all ratios of what is used for educational events like lectures and what is available in the school building.

$$Utilisation, \qquad U = \frac{used\ seat\text{-}hou}{total\ seat\text{-}hours\ available}$$

$$Frequency, \qquad F = \frac{used\ roomslots}{total\ roomslots}$$

$$Occupancy, \qquad O = \frac{used\ seat\text{-}hours\ within\ used\ roomslots}{total\ seat\text{-}hours\ available\ with\ \ \ used\ roomslots}$$

$$Relation: U = F * O$$

Consider the following example:
We have a school with 3 classrooms with a capacity of 30 seats.
That school is open 1 day a week, and that day has 2 timeslots: morning and afternoon.
There are 2 classes with 25 students which attend lectures in both the morning and the afternoon. Figure 3.1-1 depicts the usage of seats in the school. During both timeslots, one room has 30 unused seats in it while two rooms contains 25 used seats and 5 unused seats.

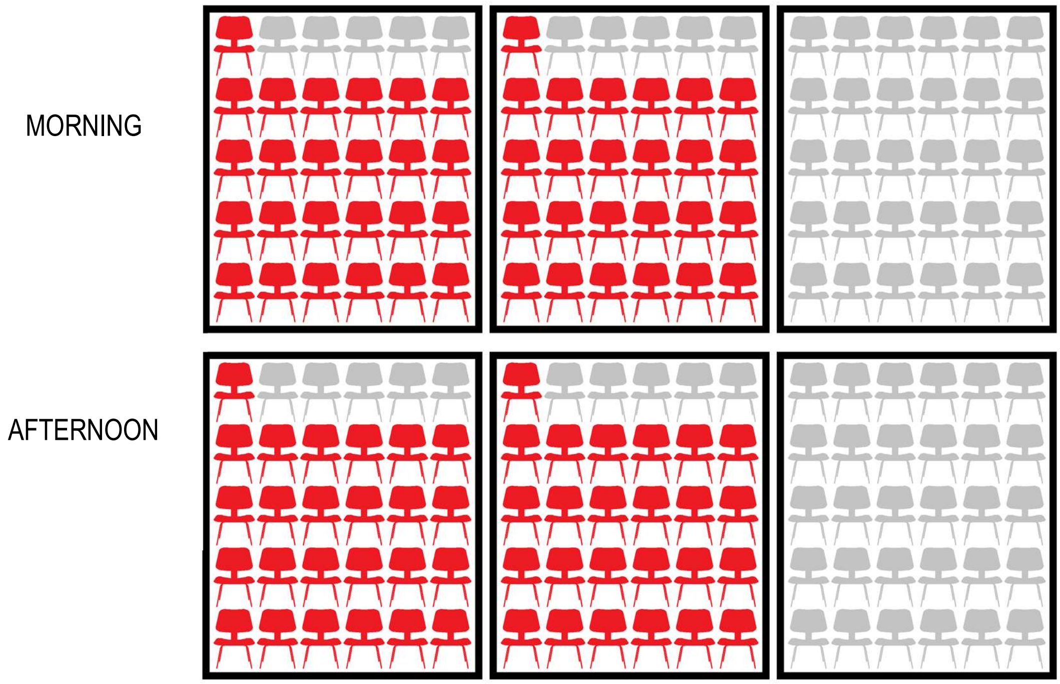


FIGURE 3.1-1 USAGE OF SEATS IN A SMALL SCHOOL

$$Utilisation, \qquad U = \frac{(2\ classes * 25\ students) * 2\ timeslots}{(3\ rooms * 30\ seats) * 2\ timeslots} = \frac{100}{180} = \frac{5}{9}$$

$$Frequency, \qquad F = \frac{2\ lectures * 2\ timeslots}{3\ rooms * 2\ timeslots} = \frac{4}{6}$$

$$Occupancy, \qquad O = \frac{(2\ classes * 25\ students) * 2\ timeslots}{(2\ rooms * 30\ seats) * 2\ timeslots} = \frac{100}{120} = \frac{5}{6}$$

$$U = F * O \qquad U = \frac{4}{6} * \frac{5}{6} = \frac{4 * 5}{6 * 6} = \frac{20}{36} = \frac{5}{9}$$

The utilisation, frequency and occupancy from the example are calculated above. They all can point to certain problems. A utilisation close to 100% indicates that all is well: rooms are often used and the class sizes are close to the room sizes. A low frequency indicates that there is a surplus of rooms as they are often not used (this can vary per room size or type). The occupancy indicates if the room sizes match the classes sizes well (high occupancy) or not (low occupancy).

In our project, we want to make full use of the resources at hand. Therefore we will focus on utilisation rather than frequency and occupancy. If utilisation is lower than expected under certain circumstances, we can look into the other two measures to determine the causes.

3.1.2 DEFINITION OF CRITICAL UTILISATION

We want to assess the interaction between demand and resource levels. One of the interests is a minimisation of costs, which can be done by minimising resources. Therefore we focus on the minimum resource level needed to meet all demand. With our model we create a scenario which consists of classes of students, the events they will attend, lecturers that teach each event and some restrictions on when or where specific events must be scheduled and determine what the minimum set of rooms is to create a feasible timetable. This is visualised in Figure 3.1-2 as communicating vessels with a liquid in them. If the number of rooms is too low, not all events, the liquid, will fit in the timetable. By trying lots of sets of rooms, we identify the lowest level of rooms that can house all events.

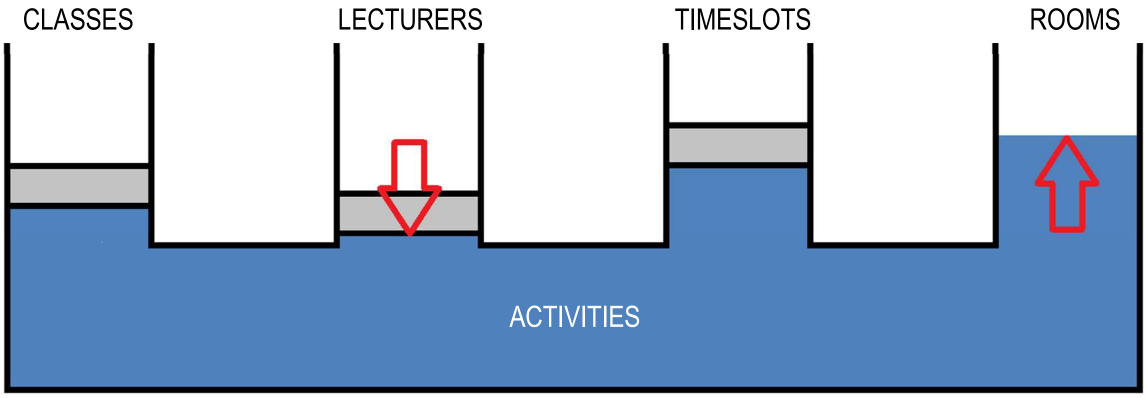


FIGURE 3.1-2 SIMPLIFIED INTERACTION OF RESOURCE LEVELS

As events vary from time to time and additional space can be acquired and abandonned, we work with a general concept: utilisation. When the staff levels, rooms etc. grow proportionally with the number of enrolled students, the utilisation of the educational space stays constant and timetables will have the same quality. Therefore one could estimate the demand for educational space based on the number of events (which depends on student numbers) and a utilisation.

We are looking for a utilisation U_C below which it is almost always possible and above which it is almost always impossible to generate a feasible timetable. We call U_C, the *critical utilisation*. This would be the utilisation in the situation that the total room capacity is precisely at the liquid level in the example above.

3.1.3 DETERMINATION OF CRITICAL UTILISATION

We use a model by Beyrouthy et al. (2006) to find the critical utilisation. This model also uses two other concepts: requested and achieved utilisation. Requested utilisation, U_R, includes all events that were requested to be included in a timetable whereas achieved utilisation, U_A, only includes events that made it into the timetable.

$$\textit{Requested utilisation,} \qquad U_R = \frac{\textit{seat-hours needed for all events}}{\textit{total seat-hours available}}$$

$$\textit{Achieved utilisation,} \qquad U_A = \frac{\textit{seat-hours of scheduled events}}{\textit{total seat-hours available}}$$

We vary the requested utilisation by using different sets of rooms to schedule the same set of events. Then we use a simulated annealing heuristic to generate a timetable that complies with all policies and can be supported by chosen resource levels. If we have enough space $U_A = U_R$ but at some point, no feasible timetable can be generated. The solution is that some events remain unscheduled and $U_A < U_R$. The utilisation where U_A starts to deviate from U_R is the critical utilisation U_C. This works the same with frequency.

Consider the example from section 3.1.1 again.
We limit ourselves to the morning: A school with 3 rooms with 30 seats and 2 classes with 25 students. Both classes take one course and we have the policy that classes may not be merged into one room.
We keep the events the same (two lectures, requiring 25 seat-hours each) and vary the set of rooms: first we can use all three rooms, then only two and finally the set is limited to one room.
There is no difference between classes or rooms, so we first schedule the lectures arbitrarily in the last two rooms as shown in Figure 3.1-3. The last room is not included in the second set of rooms, therefore we move the lectures to the first two rooms. Finally we are left with only one room so one of the lectures is not scheduled.
In the example, the critical utilisation must lie between 83.3% and 166.7% and one can deduce that $U_C = 83.3\%$ if all classes consists of 25 students and all rooms have 30 seats. However, real life situations are not so homogeneous and we are not be able to generate optimal timetables. The timetabling heuristic generates near-optimal timetables but may deviate slightly from the true optimum. Beyrouthy et al. (2008) found that when plotting U_R and U_A against each other, U_C can be identified. These graphs consist of a lot of data points that are the result of scheduling a set of events under equal circumstance in a varying set of rooms. They start out in a straight line where $U_A = U_R$ and beyond a certain point start to deviate more and more from that line, that point is U_C. Figure 3.1-4 is an example of such a graph.
Each dot represents a solution to a scheduling problem, a timetable created by running the scheduling heuristic once. The graph is the result of an experiment. The experiment in

the example consists of 45 runs, dots. Each run in an experiment is subject to the same constraints but the set of rooms in which can be scheduled varies.

$$U_R = \frac{50}{90} = 55.6\%$$

$$U_A = \frac{50}{90} = 55.6\%$$

$$U_R = \frac{50}{60} = 83.3\%$$

$$U_A = \frac{50}{60} = 83.3\%$$

$$U_R = \frac{50}{30} = 166.7\%$$

$$U_A = \frac{25}{30} = 83.3\%$$

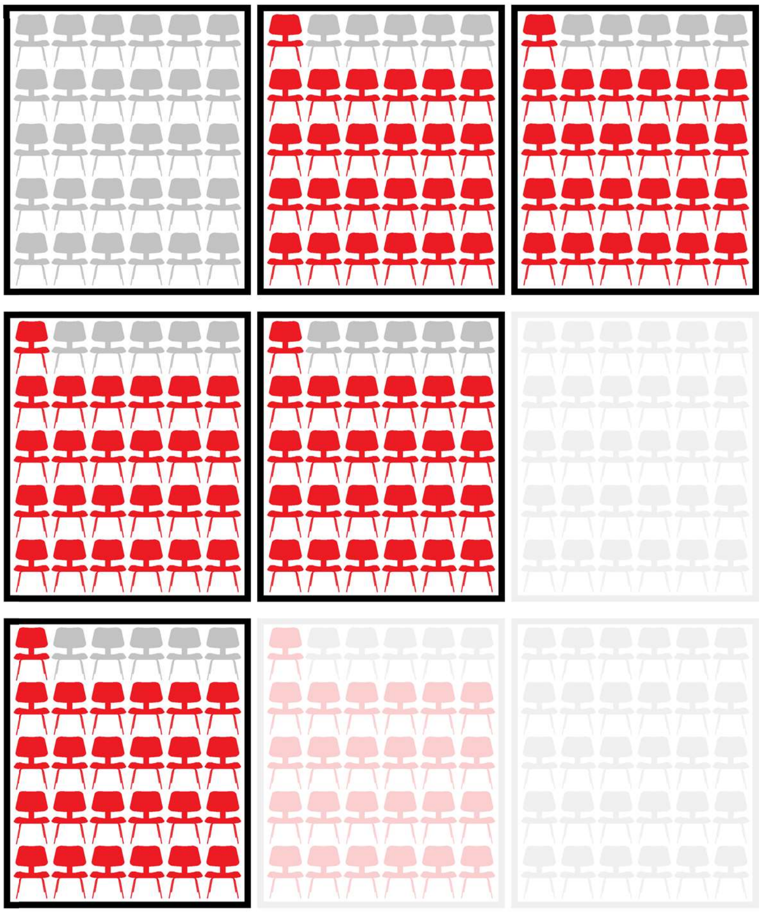


FIGURE 3.1-3 EXAMPLE OF THE CALCULATION OF REQUESTED AND ACHIEVED UTILISATION

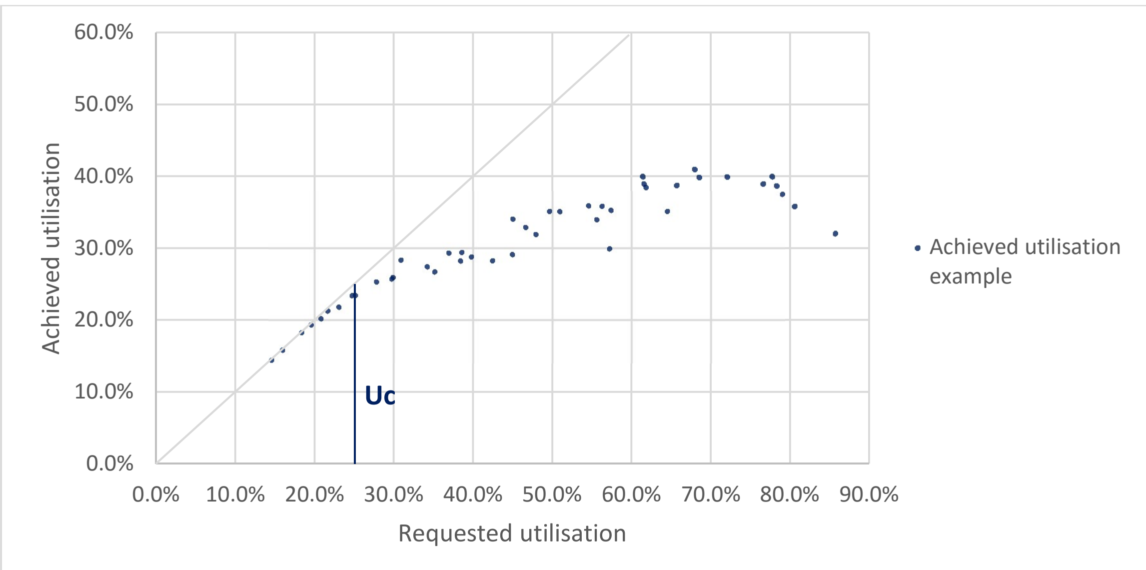


FIGURE 3.1-4 EXAMPLE OF UTILISATION GRAPH

3.1.4 APPLICATION OF THE CRITICAL UTILISATION

The critical utilisation of educational space can be used for a few purposes. One can determine if there is overcapacity by comparing the current utilisation to the U_C that is found when we use the real life resources and demand.
Another application is the comparison what resource levels are needed to get a certain U_C. This can help to decide on the trade-off between housing and staffing costs.
Furthermore, one could see how U_C would change when certain policies are adopted. This shows if they are possible within the limit of the current resources or give an indication of the increase in housing costs.
Beyrouthy et al. give a model to determine critical utilisations which can be applied to find the information we seek. We will use this model as the basis of the tool that is the objective of this problem: A tool that enables the school management to assess the impact of school policies on the timetables in general and on student satisfaction specifically.

3.2 AUTOMATED TIMETABLING

Each data point in the utilisation charts discussed in section 3.1.3 corresponds to a set of events that was scheduled, a timetable. We need a way to create timetables, preferably automatically. Timetabling is a combinatorial optimisation problem. We can combine the lectures, rooms, teaching staff and classes in a finite number of ways. That may be good news, the bad news is that the problem is too large to solve in an acceptable time by enumerating all options. Therefore we will use heuristics to find a solution that is close to optimality in an acceptable time frame.

3.2.1 SIMILAR SCHEDULING PROBLEMS

There are numerous approaches with even more varieties that can be used to construct and refine timetables. Which one works best depends strongly on the problem instance (Ahmed, Özcan, & Kheiri, 2015; Bettinelli, Cacchiani, Roberti, & Toth, 2015; Pillay, 2014). So we will try to find comparable problems in scientific literature and find the heuristic or algorithm that copes best with such problems.
We found various timetabling competitions where different teams tried to find the best timetables for a given set of events and constraints and often in limited time. The performance of approaches can be compared rather easily. Bettinelli et al. (2015) reviewed the International Timetabling Competition (ITC-2007) and work that followed from this competition. University timetabling is split into three disciplines: Examination Timetabling, Post-Enrolment Course Timetabling and Curriculum-Based Course Timetabling (CB-CTT). Ahmed et al. (2015) adds High School Timetabling (HTT) which overlaps with CB-CTT. In our problem we want to schedule series of lectures of a class of students at certain times in a room with specific lecturers. We do not deal with large fluctuations in class size and students stay in the same groups for all their courses. As we work with classes, we will not split them in smaller groups for certain events.
These characteristics fit Curriculum-Based Course Timetabling better than Post-Enrolment Course Timetabling where students can enrol in individual courses and events may be split. Examination timetabling is a bit simpler: scheduling exams such that all students fit in the exam room and do not have to be in two places at a time. Lecturers and room types are often disregarded, which is not the case in our situation. Therefore we will consider heuristics to solve CB-CTT or HTT.

3.2.2 CHOICE OF HEURISTIC

Ahmed et al. (2015) reviewed the International Timetabling Competition of 2011 and compared the heuristics that were used to solve HTT problems. First a local operator is selected to change the current solution, than a heuristic is used to accept or reject the new solution.
The heuristic uses multiple local operators, one of which could be swapping the timeslot of two randomly selected events or remove an event from the timetable and replace it with an unassigned event. There are several ways to select a local operator, from random selection or using a different one for each step in a predefined sequence up to selecting a heuristic with a probability that is based on their performance in the past.
The solution acceptance heuristics that were used are: Simulated Annealing and Great Deluge. Simulated Annealing accepts worsening solutions with a probability that lowers during the search while Great Deluge accepts worsening solutions if the deterioration is less than a threshold that lowers during the search.
The best performing approach in ITC-2011 paired Simulated Annealing with random selection of low level-heuristics. Ahmed et al. (2015) propose an approach that uses Great Deluge and alternates local operators in a sequence. Their approach works almost as well as the winner's and performs better with larger problems. Which leads to their advice to avoid probability based methods (Simulated Annealing and random selection) and use an approach based on Great Deluge.

Bettinelli et al. (2015) looked into ITC-2007 and the work on CB-CTT that followed from it. Their conclusion was that there was no timetabling heuristic that outperformed all others in all cases. Pillay (2014) examines if there is a methodology that can cope with various timetabling problems. Different benchmark problems were gathered and there was one heuristic by Bai et al. (2012) that outperformed all others on both CB-CTT benchmark problems. This heuristic uses Simulated Annealing with a stochastic heuristic selection strategy with a short-term memory, i.e. low level heuristics are selected with a probability that depends on the improvement caused in the last few instances it was used.
The benchmark problems for HTT did not present a heuristic that was clearly the best. The best ranking heuristic, by Kalender, Kheiri, Özcan, and Burke (2013), combined Simulated Annealing with greedy-gradient selection of local operators. Greedy-gradient selection tests all low level heuristic and uses that one until no further improvements are possible after which it will repeat that procedure.

The situation at the AUAS resembles CB-CTT or HTT problems that are used for benchmarking and timetabling competitions. In these competitions, Simulated Annealling based heuristics perform consistently well. We find the heuristic presented by Bai et al. (2012) most promising due to its performance compared to other heuristics and the fact that it favours local operators that perform well under the circumstances at hand. We intend to vary scheduling the problem size, resource capacities and other constraints a lot and we expect that the heuristic of Bai et al. (2012) will cope best with those variations.

3.3 TIMETABLING WITH SIMULATED ANNEALING

The timetabling method that we selected (in section 3.2.2) consists of a simulated annealing heuristic with some low level heuristics that provide input. There are various parameters that can influence the performance of the heuristic. This section explains how the heuristics works, tells what settings were used and how we came to them.

3.3.1 INTRODUCTION TO SIMULATED ANNEALING

We approach timetabling as a deterministic problem: all events are known in advance and are treated as if they are certain to take place. This leaves us with fitting all events in a timetable.

Simulated annealing is useful to find a near optimal solution for combinatorial optimisation problems. These problems have a (large but) finite number of possible solutions with one or more optimal solutions (highest profit, lowest number of penalties, shortest lead time etc.). This optimal solution can be found by trying all possibilities and checking which one is the best. When this takes too long, trying some promising solution directions and using the best solution found in these trials is a good shortcut. Which happens to be the essence of simulated annealing. The next few paragraphs explain our simulated annealing heuristic step by step.

We use a heuristic to generate a timetable that can be carried out with the resources at hand, complies with all policies and has the highest utilisation possible. In order to do so we need combine all these elements into one goal function: We award penalties based on each element and try to minimise the total penalty score.

The solution to our problem is a timetable, the assignment of all events e to a room r and timeslot t which is represented by $\boldsymbol{a}$. The solution space S is the group of all possible solutions: all possible combinations of assignments of events, $\boldsymbol{a}(e)$, to pairs (r,t). The heuristics tries to find optimal solution $\boldsymbol{a}^* \in S$.

We cannot evaluate all solutions to the timetabling problem and only want to look into promising solution directions. Therefore we evaluate solutions close to the current solution (which should improve over time) and continue from there if it is a better solution or try again. Such neighbouring solutions are often called the neighbourhood of $\boldsymbol{a}$, $N(\boldsymbol{a})$. Each iteration, we use a simple local operator to create a neighbour, a candidate solution $\tilde{\boldsymbol{a}}$, which becomes the current solution $\boldsymbol{a}$ if it is accepted.

$E = \{all\ events\}$
$R = \{all\ rooms\}$
$T = \{all\ timeslots\}$

$\boldsymbol{a}(e) = (r,t) = assignment\ of\ event\ e\ to\ room\ r\ and\ timeslot\ t$
$\boldsymbol{a} = solution\ to\ the\ timetabling\ problem, consisting\ of\ all\ event\ assignments\ \boldsymbol{a}(e)$
$S = Solution\ space, set\ of\ all\ possible\ solutions\ \boldsymbol{a}$

We cannot evaluate all solutions to the timetabling problem and only want to look into promising solution directions. Therefore we evaluate solutions close to the current solution (which should improve over time) and continue from there if it is a better solution or try again. Such neighbouring solutions are often called the neighbourhood of $\boldsymbol{a}$, $N(\boldsymbol{a})$. Each iteration, we use a simple local operator to create a neighbour, a candidate solution $\tilde{\boldsymbol{a}}$, which becomes the current solution $\boldsymbol{a}$ if it is accepted.

$N(\boldsymbol{a}) = neighbourhood\ of\ \boldsymbol{a}$
$N(\boldsymbol{a}) \subseteq S$
$\tilde{\boldsymbol{a}} \in N(\boldsymbol{a})$

Whether or not a solution is accepted depends on the penalty score, which is a function of the solution. The acceptance criteria and penalty function are described in one of the following sections, just as the working of the heuristic and the local operators.

$\min penalty\ score = f(\boldsymbol{a})$

3.3.2 ITERATIVE PROCEDURE

During a run of a set number of iterations ($iteration_{max}$), alternative solutions are generated by changing the assignments of some events by using local operators. These are accepted if they are an improvement, in our case when they have a lower penalty score, or by chance (based on the size of the deterioration δ and the temperature $t_{current}$). To determine if a solution with a higher penalty score is accepted, an acceptance function is used. The procedure is given in pseudo code below.

$$\delta = f(\tilde{\boldsymbol{a}}) - f(\boldsymbol{a})$$

$$Accept\ solution\ if: \quad \delta < 0$$

$$or\ if: \quad X < e^{-\left(\frac{\delta}{t_{current}}\right)}, \ X \sim U(0,1)$$

```
while (iteration_c < iteration_max)
        select a local operator
        generate candidate solution ã
        if δ <= 0 and a new solution was generated then
                a = ã
        endif
        if δ > 0 and  f (δ,t_current) > random(0,1) then
                a = ã
        endif
        adjust t_current
        iteration_c =  iteration_c + 1
loop
```

3.3.3 ACCEPTANCE FUNCTION

When a candidate solution is worse than the current solution, it is only accepted with a certain probability. In our model a random number between 0 and 1 (named X above) is generated, if it is lower than $e^{(-\delta/t)}$ the candidate solution is accepted.

$$Acceptance\ probability = \begin{cases} e^{\frac{-\delta}{t_{current}}} & if\ \delta > 0 \\ 1 & if\ \delta \leq 0 \end{cases}$$

This acceptance function was part of the model proposed by Bai et al. (2012). Others have had success with different acceptance criteria. By dividing the deterioration of the solution value by a power of the temperature, the acceptance probabilities become more extreme, a lot higher for $t > 1$ and a lot lower for $t < 1$, see Figure 3.3-1.
However, at the acceptance probability at temperatures below one is almost zero with the penalty values we use. Hence, the practical difference is that a wider range of solutions can be explored early on in the annealing process.

We have tried a nine runs at two temperature settings with the following acceptance criteria:

$$Acceptance\ probability = \begin{cases} e^{\frac{-\delta}{t_{current}{}^2}} & if\ \delta > 0 \\ 1 & if\ \delta \leq 0 \end{cases}$$

$$Acceptance\ probability \quad \begin{cases} e^{\frac{-\delta}{t_{current}{}^3}} & if\ \delta > 0 \\ 1 & if\ \delta \leq 0 \end{cases}$$

Results of this comparison are shown in Figure 3.3-2. The runs with t_{start} = 1000 and the original acceptance criterion perform worse than the others, which perform very similar. Therefore we choose not to deviate from Bai et al. (2012) and use the original acceptance function.

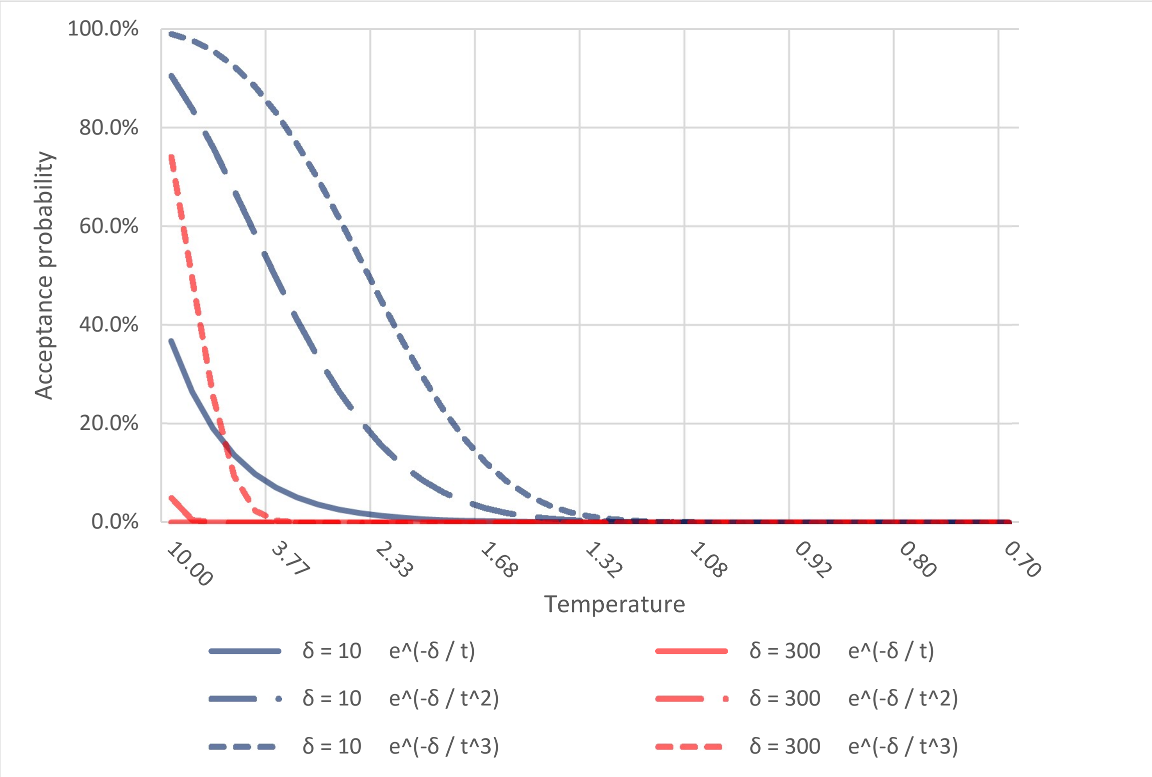


FIGURE 3.3-1 ACCEPTANCE PROBABILITIES OF CANDIDATE SOLUTIONS (WITH EITHER δ=10 OR 300) WITH A DECREASING TEMPERATURE.

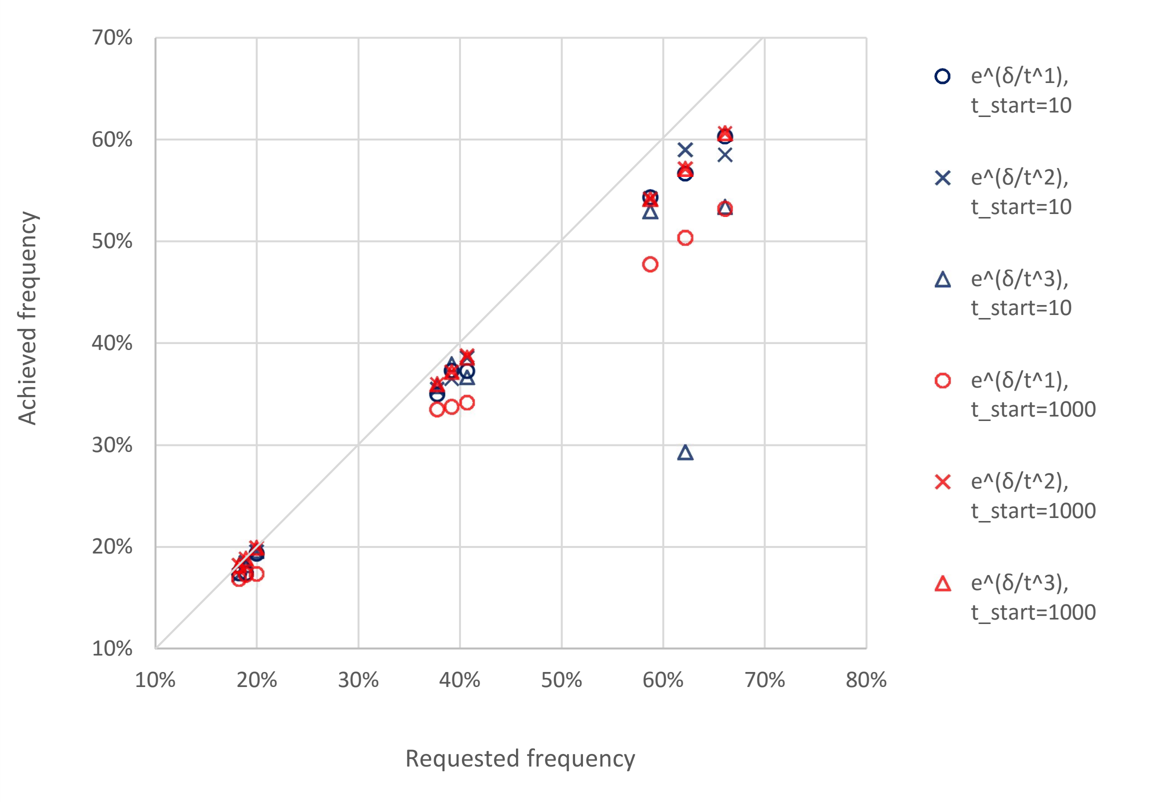


FIGURE 3.3-2 COMPARISON OF THE PERFORMANCE OF DIFFERENT ACCEPTANCE CRITERIA

3.3.4 TEMPERATURE ADJUSTMENT

The temperature is decreased after each period of n iterations. Temperature t is changed into new temperature as described below. Cooling factor β is a constant, chosen such that the temperature decreases from start to end temperature t_{st} and t_{end} in the predetermined maximum number of iterations.

$$After\ n\ iterations: \quad t_{current} := \frac{t_{current}}{1 + \beta * t_{current}}$$

$$with: \quad \beta = \frac{(t_{st} - t_{end}) * n}{t_{st} * t_{end} * iteration_{max}}$$

However the model can get stuck in a local optimum, making it difficult to find new, acceptable solutions in the search for the optimal solution. Whenever fewer solutions (both better and worse) are accepted during a learning period of lp iterations (which can be but does not necessarily have to be n) than predetermined minimal acceptance ratio r_{min}, we start reheating. During reheating, the temperature increases each iteration (which is much faster than the cooling pace). To do so, we must save the temperature each time an improved solution is found: $t_{impr} := t_{current}$. We use this as a starting point when we increase the temperature each iteration in the way shown below:

$$When\ reheatin\ : \quad t_{impr} := \frac{t_{impr}}{1 - \beta * t_{impr}}$$

$$t_{current} := t_{impr}$$

The iterative procedure of the heuristic is expanded with the adjustment of the temperature in the pseudo code below. An example of how the temperature is adjusted during a run is given afterwards.

```
while (iteration_c < iteration_max)
    select a local operator
    generate candidate solution ã
    if δ <= 0 and a new solution was generated then
        a := ã
        t_impr := t_current
        stop reheating
    endif
    if δ > 0 and  f (δ,t_current) > random(0,1) then
        a := ã
    endif
    if reheating then
        t_impr := t_impr / (1 - β * t_impr)
        t_current := t_impr
    endif
    if mod(iteration_c, lp) = 0 then
        t_current := t_current / (1 + β * t_current)

        if ((# accepted ã) / lp) < r_min then
            start reheating
            t_impr := t_impr / (1 - β * t_impr)
            t_current := t_impr
        else
            t_current := t_current / (1 + β * t_current)
        endif
    endif
    iteration_c :=  iteration_c + 1
loop
```

We start the reheating process with the best solution found so far (which is not necessarily the current solution). In the example below, $r_{min} = 0.5$, we start at $t_{current} = 0.8$ and decrease the temperature every 3 iterations and $lp = 6$. So at least half, three out of six, candidates must be accepted during each learning period, otherwise we will start reheating.

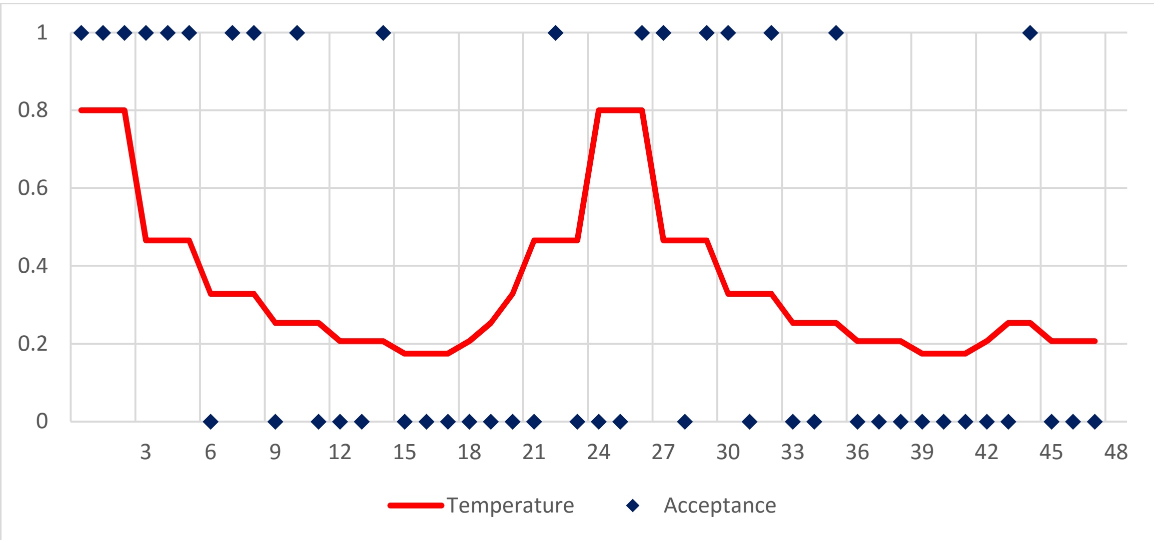


FIGURE 3.3-3 EXAMPLE OF TEMPERATURE DEVELOPMENT DURING SIMULATED ANNEALING

Almost all solutions are accepted in the first iterations (acceptance = 1) so the temperature drops steadily until iteration 18. (Iteration 6 was the first one that was not accepted: acceptance = 0.) Between iteration 12 and 18, only 1/6 solutions is accepted and as $1/6 < r_e$, we start reheating and the temperature rises quickly until an improved solution is found in iteration 22. Reheating is stopped and the temperature will drop every three iterations. However, between iteration 18 and 24 only one solution was accepted and reheating starts again. The temperature is capped at the starting temperature and stays stable until the acceptance of the solution in iteration 26 ends the reheating phase, letting the temperature drop again before iteration 27, 30, 33, 36 and 39. The lack of accepted solutions between iteration 36 and 42 starts the last reheating phase.

3.3.5 CANDIDATE SOLUTIONS AND LOCAL OPERATORS.

In our model we use seven different local operators to alter our solution. These simple heuristics are taken from both Bai et al. (2012) and Beyrouthy et al. (2008). They cover various swaps of assignments between events and un-allocating allocated events or allocating unallocated events. (Allocated events are included in the timetable where unallocated events are not.)

Neighbour solutions of solution $\boldsymbol{a}$ are a subset of the total solution space. Those neighbours $\tilde{\boldsymbol{a}}$ are the same as $\boldsymbol{a}$ except for the assignments of a few events. The section below gives an example and description of all seven local operators, o. Figure 3.3-4 depicts these examples.

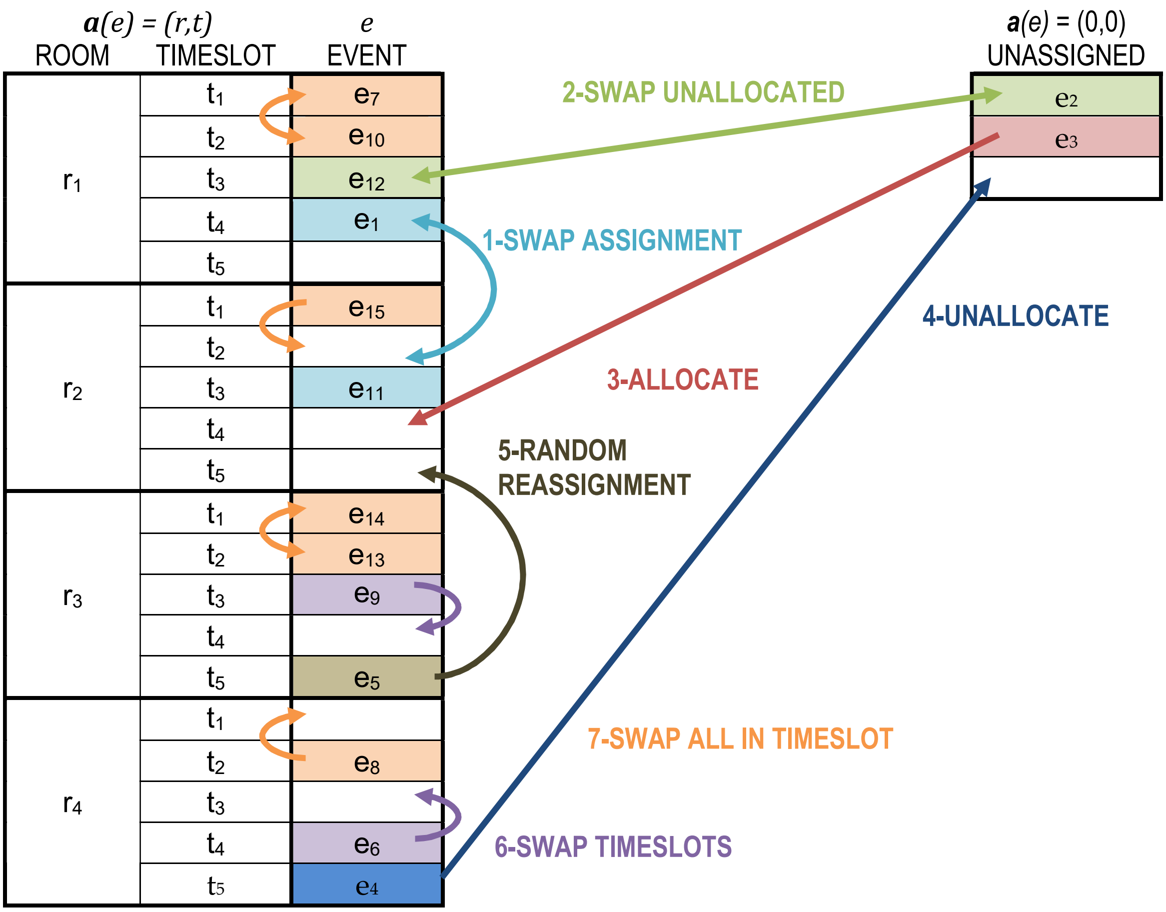


FIGURE 3.3-4 VISUALISATION OF POSSIBLE SWAPS BY LOCAL OPERATORS

$N(\boldsymbol{a}) \subseteq S$
$\tilde{\boldsymbol{a}} \in N(\boldsymbol{a})$
$E \ = \ \{all\ events\} \quad = \{e_1, e_2, e_3 \dots e_{15}\}$
$R \ = \ \{all\ rooms\} \quad = \{r_1, r_2, r_3, r_4\}$
$T \ = \ \{all\ timeslots\} \ = \ \{t_1, t_2, t_3, t_4, t_5\}$

Operator 1 – Swap two
Two rooms are randomly selected and in each room an event is randomly chosen. The rooms and timeslots of these events are swapped. In case we select room r_1 and r_2 and then select two events, e_5 and e_7, at timeslots t_{11} and t_{12} respectively, $\tilde{\boldsymbol{a}}$ is the same as $\boldsymbol{a}$ except for $\boldsymbol{a}(e_5)$ and $\boldsymbol{a}(e_7)$.
$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) = (r, t | r = r_1) \ \rightarrow \ \boldsymbol{a}(e_a) = \boldsymbol{a}(e_1) = (r_1, t_4)$
$\boldsymbol{a}(e_b) \in \boldsymbol{a}(e) = (r, t | r = r_2) \ \rightarrow \ \boldsymbol{a}(e_b) = \boldsymbol{a}(e_{11}) = (r_2, t_3)$

$\tilde{\boldsymbol{a}}(e_5) = (r_2, t_3)$
$\tilde{\boldsymbol{a}}(e_7) = (r_1, t_4)$

Operator 2 – Swap unallocated
One of the unallocated and one allocated events are randomly selected and swapped. The room and timeslot remain unaltered.
$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) \neq (0,0) \ \rightarrow \ \boldsymbol{a}(e_a) = \boldsymbol{a}(e_{12}) = (r_1, t_3)$
$\boldsymbol{a}(e_b) \in \boldsymbol{a}(e) = (0,0) \ \rightarrow \ \boldsymbol{a}(e_a) = \boldsymbol{a}(e_2)$

$\tilde{\boldsymbol{a}}(e_5) = (0,0)$
$\tilde{\boldsymbol{a}}(e_6) = (r_1, t_3)$

Operator 3 – Allocate
An unallocated event, a room and timeslot are randomly selected. The event is added to the timetable on that time and place.
$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) = (0,0) \ \rightarrow \ \boldsymbol{a}(e_a) = \boldsymbol{a}(e_3)$
$(r |\ r \neq 0) \ \rightarrow \ r_2$
$(t |\ t \neq 0) \ \rightarrow \ t_5$

$\tilde{\boldsymbol{a}}(e_a) = (r_2, t_5)$

Operator 4 - Unallocate
Un-allocate a randomly selected allocated event.
$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) \neq (0,0) \ \rightarrow \ \boldsymbol{a}(e_a) = \boldsymbol{a}(e_4) = (r_4, t_5)$

$\tilde{\boldsymbol{a}}(e_7) = (0,0)$

Operator 5 – Random reassignment
Select a random event and schedule it on a randomly selected place and time.
$e \in E \ \rightarrow \ e = 5$
$(r |\ r \neq 0) \ \rightarrow \ r_2$
$(t |\ t \neq 0) \ \rightarrow \ t_4$

$\tilde{\boldsymbol{a}}(e_5) = (r_2, t_4)$

Operator 6 – Swap timeslot of two
Select randomly select two events and swap their timeslots (and room if one of the events is not assigned to a timeslot).

$e_a \in E \ \ \rightarrow \ \ e = 6$
$e_b \in E \ \ \rightarrow \ \ e = 9$
$\boldsymbol{a}(e_6) = (r_4, t_4)$
$\boldsymbol{a}(e_9) = (r_3, t_3)$

$\tilde{\boldsymbol{a}}(e_6) = (r_4, t_3)$
$\tilde{\boldsymbol{a}}(e_9) = (r_3, t_4)$
$e_a \in E \ \ \rightarrow \ \ e = 6$
$e_b \in E \ \ \rightarrow \ \ e = 3$
$\boldsymbol{a}(e_6) = (r_4, t_4)$
$\boldsymbol{a}(e_3) = (0,0)$

$\tilde{\boldsymbol{a}}(e_6) = (0,0)$
$\tilde{\boldsymbol{a}}(e_3) = (r_4, t_4)$

Operator 7 – Swap all in timeslots
Select two timeslots (t_1 and t_2) and swap the timeslots of all events assigned to either timeslot.

$\boldsymbol{a}(e_a) \forall \boldsymbol{a}(e) = (r, t | t = t_1)$
$\boldsymbol{a}(e_b) \forall \boldsymbol{a}(e) = (r, t | t = t_2)$

$\tilde{\boldsymbol{a}}(e_a) = (r, t_2)$
$\tilde{\boldsymbol{a}}(e_b) = (r, t_1)$

In case an assignment is impossible (the kind of events needed for a swap do not exist, or an event that takes multiple timeslots is split over multiple days), the reassignment is cancelled. Otherwise it is marked as a new candidate solution and it is either accepted or rejected. The current solution is not changed when a candidate solution $\tilde{\boldsymbol{a}}$ is not accepted.

3.3.6 OPERATOR SELECTION

We choose a local operator from among the ones mentioned in the previous paragraph by chance each iteration. The probability of selecting and operator *o* is based on a weight factor, w_o. As some operators will perform better in certain phases of the simulated annealing process than others, we vary the probability of each heuristic during a run based on their performance in the previous iterations. In order to do so we count various things after each iteration, that information is used after each learning period to update the weight factors of all operators. We keep track of the following things:

c_{tot_o}	The total number of times operator *o* is selected.
c_{new_o}	The number of times operator *o* generates a new and feasible solution.
c_{acc_o}	The number of times a solution generated with operator *o* is accepted.
c_{acc_all}	The total number of solutions that is accepted (all operators combined).

After each learning period we evaluate the counters and decide if we start a reheating phase and update the probabilities with which we select each local operator. These probabilities depend on weight factors w_o. There is a minimum weight factor w_{min} that ensures that all heuristics have a chance to be selected every time. The selection probability of operator o, pr_o is calculated as shown below. When we give the heuristics subsequent intervals with a width of pr_o, we can generate a random number between 0 and 1 and the heuristic that corresponds to the interval in which the random number lies is selected.

$$pr_o = \frac{w_o}{\sum_O w_o}$$

$$Select\ operator\ i\ when{:}\quad \sum_{o=1}^{i-1} pr_o < X < \sum_{o=1}^{i} pr_o \qquad X{\sim}U(0,1)$$

The weight factors and selection probabilities are updated after every lp iterations. When the fraction of solutions that was accepted in the past lp iterations is larger than acceptance ratio r_{min} the weight factor of an operator is the ratio between the number of accepted solutions created by operator o and the number of times o was used.
However, that acceptance rate is not too important when not enough candidate solutions are accepted. In that case, operators that create new, feasible solutions are most promising. Therefore the new weight factors are in that case equal to the ratio of new solutions created with o and the number of times o was selected.
The weight factors should be at least w_{min} so if the calculations mentioned above cannot be done because an operator was not used during the learning period or have an outcome that is too low than $w_o = w_{min}$.. The pseudo code below includes the operator selection.

```
while (iteration_c < iteration_max)
        iteration_c = iteration_c + 1
        pr = 0
        for ∀o
                pr = pr + pr_o
                if random(0,1) < pr then
                        generate ã with operator o
                        c_tot_o     = c_tot_o + 1
                        exit for-loop
                endif
        next
        if δ <= 0 and a new solution was generated then
                a = ã
                t_impr = t_current
                stop reheating
                c_new_o  = c_new_o + 1
                c_acc_o  = c_acc_o + 1
                c_acc_all = c_acc_all + 1
        endif
        if δ > 0 and a new solution was generated then
                c_new_o  = c_new_o + 1
                if f (δ,t_current) > random(0,1) then
                        a = ã
                        c_acc_o   = c_acc_o + 1
                        c_acc_all = c_acc_all + 1
                endif
```

```
endif
if reheating then
        t_impr := t_impr / (1 - β * t_impr)
        t_current := t_impr
endif
if mod(iteration_c, lp) = 0 then
        if (c_acc_all / lp) < r_min then
                start reheating
                t_impr := t_impr / (1 - β * t_impr)
                t_current := t_impr
                for ∀o
                        if c_total_o = 0
                                w_o = w_min
                        else
                                w_o = max{w_min, c_new_o / c_acc_all}
                        endif
                next

        else
                t_current := t_current / (1 + β * t_current)
                for ∀o
                        if c_total_o = 0
                                w_o = w_min
                        else
                                w_o = max{w_min, c_acc_o / c_acc_all}
                        endif
                next
        endif
        c_tot_o = 0
        c_new_o = 0
        c_acc_o = 0
        c_acc_all = 0
endif
loop
```

3.3.7 PENALTY STRUCTURE

All penalties are split into three levels. Hard and resource constraints are in the top level, followed by underutilisation in the middle and finally soft constraints in the lowest level. Underutilisation will be measured in seat-hours and an average class consists of around 25 students. With that in mind we set the penalties at 1000 and 1 for top and low level constraint violations and either 10 points per unused seat-hour (25 * 10 = 250) or 250 points for an unused roomslot. These amounts ensure that the total amount of possible penalties at a level do not exceed a single penalty at a higher level and that we first solve all top level constraint violations, than all mid level constraint violations and finally, if possible low level constraint violations.

A basic hard constraint is: Each class can only attend one event per timeslot. A soft constraint, often a preference, could be: All lectures should take place between 10:00 and 15:00. If two lectures of the same class are both scheduled at 10:00, moving one of them to a timeslot after 15:00 causes a 1 point penalty while reducing the penalty score with 1000 for a total improvement of 999 points.

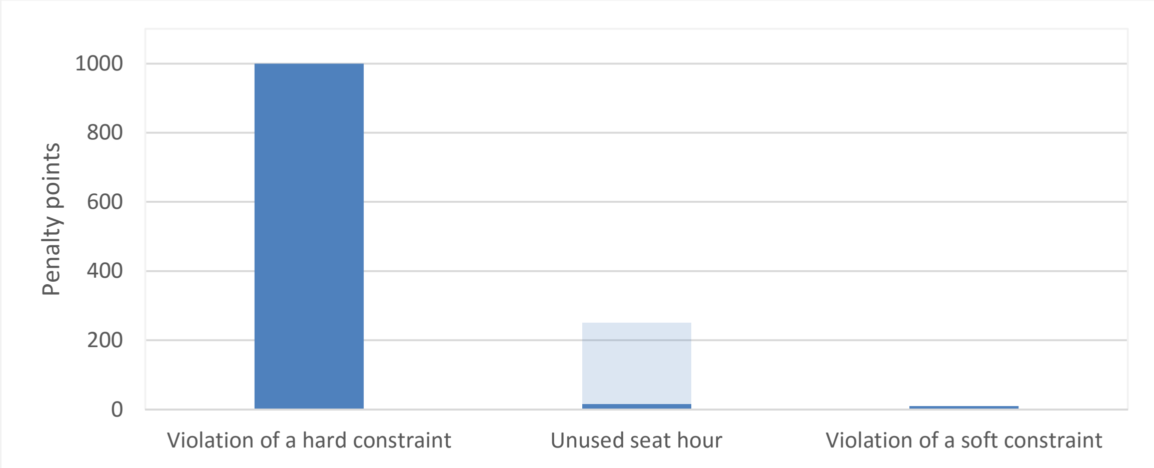


FIGURE 3.3-5 PENALTIES PER VIOLATION

Penalty score per soft constraint violation can be increased slightly to favour some violations over others. However the score per violation will not exceed 10 points.
The first 11 constraints are generally hard constraints whereas constraints 12 to 21 are soft constraints, evaluated per class. Constraint 1 through 5 are primarily related to the rooms, 6 through 9 to lecturers and 10 through 21 are class related.
Room related penalties are aimed at using as much of the available rooms as possible while making sure that the size and type are suitable for the events scheduled in them. Staff related penalties help to avoid scheduling conflicts and to comply with labour regulations. Finally the class related penalties encourage timetables that take into account the preferences of students and staff. These preferences should improve student satisfaction and study success.

$E \;=\; \{all\ events\}$
$R \;=\; \{all\ rooms\}$
$T \;=\; \{all\ timeslots\}$
$C \;=\; \{all\ classes\}$
$L \;=\; \{all\ lecturers\}$

$D \;=\; \{all\ workdays\ of\ the\ week\} = \{monday, tuesday \dots\ friday\}$
$d = 1 = \{t_1, t_2 \dots\ t_{12}\}$
$d = 2 \;= \{t_{13}, t_{14} \dots\ t_{24}\}$
$d = 3 = \{t_{25}, t_{26} \dots\ t_{36}\}$
$d = 4 \;= \{t_{37}, t_{38} \dots\ t_{48}\}$
$d = 5 = \{t_{49}, t_{50} \dots\ t_{60}\}$

$lunch(d) = \{timeslots\ that\ can\ be\ used\ as\ lunch\ break\}$
$lunch(1) = \{t_6, t_7, t_8\}$
$lunch(2) = \{t_{18}, t_{19}, t_{20}\}$
$lunch(3) = \{t_{30}, t_{31}, t_{32}\}$
$lunch(4) = \{t_{42}, t_{43}, t_{44}\}$
$lunch(5) = \{t_{54}, t_{55}, t_{56}\}$

$\boldsymbol{a} = solution\ to\ the\ timetabling\ problem$

$\boldsymbol{a}(e) = (r, t) = assignment\ of\ event\ e\ to\ room\ r\ and\ timeslot\ t$
$\pi_1(\boldsymbol{a}(e)) = r = room\ in\ the\ assignment\ of\ e$
$\pi_2(\boldsymbol{a}(e)) = t = timeslot\ \ in\ the\ assignment\ of\ e$

$N(\boldsymbol{a}) = neighbourhood\ of\ \boldsymbol{a}$
$N(\boldsymbol{a}) \subseteq S$
$\widetilde{\boldsymbol{a}} \in N(\boldsymbol{a})$

$p_i\ =\ score\ per\ penalty\ i$
$n_i\ =\ number\ of\ penalties\ i$

$$f(\boldsymbol{a}(e)) = \sum_{i=1}^{15} p_i * n_i(\boldsymbol{a}(e))$$

$$\delta_{x,y} = 1 \quad if\ x = y$$
$$\quad = 0 \quad if\ x \neq y$$

Room related penalties

Constraint 1: Rooms are used for at most one event per timeslot.

Number of penalties: The number of events in a room during a timeslot minus one (with a minimum of zero), summed over all room and timeslot combinations.

$$p_1 * n_1 = p_1 * \sum_R \sum_T \max\left(0\,, \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t}\ - 1\right)$$

Constraint 2: Rooms are large enough to seat all attending students.

Number of penalties: The total number of timeslots to which events are assigned, in a room with a capacity lower than the number of attendees.

$$p_2 * n_2 = p_2 * \sum_E \sum_R \sum_T \sum_{i=0}^{group\ size(e)-1} \left(\delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t}\ * \delta_{room\ size(r),i}\right)$$

Constraint 3: Events are scheduled in a room of a suitable type.

We use a penalty matrix $\boldsymbol{P}$ for penalty 3 to determine the size of a penalty as the severity of mismatches can vary between type combinations. (E.g. type 1 is a basic lecture room while type 2 indicates the presence of audio equipment. Events of type 1, that require only a lecture room can be assigned to type 2 rooms at a lower penalty (0.2) than the other way around, as the presence of equipment does not hinder the teaching. All other type mismatches receive a full penalty while unassigned rooms are never penalised.)

$$\boldsymbol{P} = penalty\ of\ assigning\ an\ event\ of\ type\ te\ to\ a\ room\ of\ type\ tr = \begin{bmatrix} p_{1,0} & \cdots & p_{te,0} \\ p_{1,1} & \cdots & p_{te,1} \\ \vdots & \ddots & \vdots \\ p_{1,tr} & \cdots & p_{te,tr} \end{bmatrix}$$

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0.2 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$$p_3 * n_3 = p_3 * \sum_E \sum_R \boldsymbol{P}(type(e), type(r) * \delta_{\pi_1(\boldsymbol{a}(e)),r}$$

Constraint 4: Each room is used during each timeslot.
Number of penalties: One for every room and timeslot combination with no events assigned to it (multiple events assigned to it do not cause a negative number of penalties.

$$p_4 * n_4 = p_4 * \sum_R \sum_T \max\left(0\,, 1 - \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t}\right)$$

Constraint 5: All seats in all rooms are used during each timeslot. (All seat-hours are used.)
Number of penalties: The room capacity minus the group sizes of events assigned to it with a minimum of zero, summed over all timeslots.

$$p_5 * n_5 = p_5 * \sum_R \sum_T \max\left(0, capacity(r) - \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} * group\ size(e)\right)$$

Staff related penalties

Constraint 6: Lecturers teach at most one event per timeslot.
Number of penalties: The number of events in a lecturer teaches during a timeslot beyond the first, summed over all lecturers and timeslots.

$$p_6 * n_6 = p_6 * \sum_L \sum_T \max\left(0\,, \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l} - 1\right)$$

Constraint 7: Lecturers do not teach during timeslots during which they are not available.
Number of penalties: The number of timeslots per lecturer, with events that he or she teaches assigned to it whilst being unavailable. We keep track of unavailability of all lecturers during each timeslot in matrix $\boldsymbol{U}$:

$$\begin{aligned} \boldsymbol{U}(l,t) &= 1 \quad if\ lecturer\ l\ is\ not\ available\ during\ timeslot\ t \\ &= 0 \quad if\ lecturer\ l\ is\ available\ during\ timeslot\ t \end{aligned}$$

$$p_7 * n_7 = p_7 * \sum_T \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \boldsymbol{U}(l(e), t)$$

Constraint 8: Lecturers have at least one free timeslot (to lunch) among the three lunch timeslots every day.
Number of penalties: The number of days per week that a lecturer teaches during the whole lunch period, summed over all lecturers.

$$p_8 * n_8 = p_8 * \sum_L \sum_D \delta_{\sum_{t \in lunc\ (d)} 1 - \delta_{\left(\sum_E\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}, 3}$$

Constraint 9: The workday of a lecturer does not span more than ten hours.
Number of a penalties: The number of days that the first and last event on a day that a lecturer teaches lay more than 10 timeslots apart, summed over all lecturers.
(The latest timeslot an event must be taught minus the timeslots before the first event gives the span of the day.)

$$p_9 * n_9 = p_9 * n_9 = \sum_{L}\sum_{D}\sum_{i=4+1}^{max\,hrs\,per\,day} \delta_{x,i}$$

$With$:

$$x = \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}\right)\right\} - \max\left\{(first\ timeslot\ on\ d) - 1\,, z * \left(\delta_{\left(\sum_E \sum_{t=fir}^{z}\ on\ d\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}\right)\right\}$$

$y = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$
$z = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

Class related penalties

Constraint 10: Classes attend at most one event per timeslot.
Number of penalties: The number of events in a class attends during a timeslot beyond the first, summed over all classes and timeslots.

$$p_{10} * n_{10} = p_{10} * \sum_{C}\sum_{T} \max\left(0\,, \sum_{E} \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c} - 1\right)$$

Constraint 11: Classes only suffer a low amount of low level penalties.
Number of penalties: The sum of low level penalty points awarded to a class above a threshold e.g. 20. Low level penalties have a penalty score per penalty of at most 10 points.

$$p_{11} * n_{11} = p_{11} * \sum_{C} \max\left(0\,, \sum_{i=12}^{21}\sum_{j=1}^{10}\left(p_i * n_i(c) * \delta_{p_i,j}\right) - 20\right)$$

Constraint 12: Classes have at least one free timeslot during the lunch period.
Number of penalties: The number of days per week that a lecturer teaches during the whole lunch period, summed over all lecturers.

$$p_{12} * n_{12} = p_{12} * \sum_{C}\sum_{D} \delta_{\sum_{t\in lunch(d)} 1 - \delta_{\left(\sum_E\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0},3}$$

Constraint 13: Events of a class on any given day do not span more than 8 timeslots.
Number of a penalties: The number of days that the first and last event on a day that a class must attend lay more than 8 timeslots apart, summed over all classes.
(The latest timeslot an event must be attended minus the timeslots before the first event gives the span of the day.)

$$p_{13} * n_{13} = p_{13} * \sum_{C}\sum_{D}\sum_{i=8+1}^{max\,h\quad per\,day} \delta_{x,i}$$

$With$:

$$x = \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\} - \max\left\{(first\ timeslot\ on\ d) - 1\,, z * \left(\delta_{\left(\sum_E \sum_{t=fir}^{z}\ on\ d\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$

$y = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$
$z = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

Constraint 14: A class has either no or at least four timeslots with events it will attend assigned to them.

Number of penalties: One per class, per day it has between one and three timeslots with events it must attend assigned to them.

$$p_{14} * n_{14} = p_{14} * \sum_{C}\sum_{D}\sum_{i=1}^{min\,hr\ \ per\,day-1} \delta_{(12-x),i}$$

$$x = \sum_{t=1}^{12} \delta_{\left(\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right),0}$$

Constraint 15: Events are scheduled between a predefined start and end timeslot each day.
Number of penalties: The difference between the actual and predefined start time plus the difference between the actual and predefined end time. Summed over all days and all classes.

$$n_{15} = \sum_{C}\sum_{D}(\max\{0, intended\ start\ time\ (d) - actual\ start\ time(d)\} + \max\{0, actual\ end\ time(d) - intended\ end\ time(d)\})$$

$$p_{15} * n_{15} = p_{15} * \sum_{C}\sum_{D}\left(\max\left\{0, \text{intended}\ start\ time\ (d) - \max\left\{first\ timeslot\ on\ d, 1 + x * \delta_{\left(\sum_E \sum_{t=firs\ \ on\ d}^{x}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right\}\right\} + \max\left\{0, \max\left\{y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\} - intended\ end\ time(d)\right\}\right)$$

$With$:
$x = \{first\ timeslot\ on\ day\ d, \dots, last\ timeslot\ on\ day\ d\}$
$y = \{first\ timeslot\ on\ day\ d, \dots, last\ timeslot\ on\ day\ d\}$

Constraint 16: Events that morning classes must attend are assigned to the first seven timeslots each day.
Number of penalties: For every day and each class the difference between the last timeslot an event that a class attends is assigned to and the seventh timeslot of the day.

$$p_{16} * n_{16} = p_{16} * \sum_{C \in morning\ group}\sum_{D} \max\left\{0, \max\left\{x * \left(1 - \delta_{\left(\sum_E \sum_{t=x}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\} - seventh\ timeslot(d)\right\}$$

$With$:
$x = a\ timeslot\ between\ the\ seventh\ and\ the\ last\ timeslot\ on\ day\ d.$

Constraint 17: Events that afternoon classes must attend are assigned at the earliest to the seventh timeslot of each day.
Number of penalties: For every day and each class the difference between the first timeslot an event that a class attends is assigned to and the seventh timeslot of the day.

$$p_{17} * n_{17} = p_{17} * \sum_{C \in morning\ group} \sum_{D} \left(\max\left\{0, seventh\ timeslot\ on\ d \right.\right.$$
$$\left.\left. - \max\left\{first\ timeslot\ on\ d, 1 + x * \left(\delta_{\left(\sum_E \sum_{t=fir\quad on\ d}^{x}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}\right\}\right)$$

$With$:
$x = a\ timeslot\ between\ the\ first\ and\ the\ seventh\ timeslot\ on\ day\ d.$

Constraint 18: Each class attends at least one event on Monday and Friday.
Number of penalties: One per class except for classes that have events they must attend assigned to timeslots on both Monday and Friday.

$$p_{18} * n_{18} = p_{18} * \sum_{C} 1 - \left(1 - \delta_{\left(\sum_E \sum_{t \in d = monday}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)$$
$$* \left(1 - \delta_{\left(\sum_E \sum_{u \in d = friday}\left(\delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c}\right)\right),0}\right)$$

Constraint 19: Every class attends events on a predefined number of days.
Number of penalties: One per class that has no events assigned to timeslots on the predefined number of days.

$$p_{19} * n_{19} = p_{19} * \sum_{C} \left(1 - \delta_{\left(\sum_D \left(1 - \delta_{\left(\sum_E \sum_{t \in d} \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right),0}\right)\right), predefined\ number\ of\ days} \right)$$

Constraint 20: Classes have no free timeslots between the first and last timeslot on a day during which they must attend events (disregarding any free timeslots in the lunch period).
Number of penalties: One for each free timeslot between the first and last timeslot with events for a class assigned to them each day, except for free timeslots during the lunch periods.

$$p_{20} * n_{20} = p_{20} * \sum_{C} \sum_{D} \left(\sum_{t=x}^{\min(lunc\quad(d))-1} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c},0} \right.$$
$$\left. + \sum_{u=1+\mathrm{m}\quad(lunc\quad(d))}^{y} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c},0} \right)$$

$$x = \max\left\{(first\ timeslot\ on\ d), 1 + z * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{z}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$
$$y = \max\left\{ z * \left(1 - \delta_{\left(\sum_E \sum_{t=z}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$
$$z = \{first\ timeslot\ on\ day\ d, \dots, last\ timeslot\ on\ day\ d\}$$

Constraint 21: Free timeslots during the day should be joint together. (We penalise each new series of free timeslots. I.e. the first free timeslot in a series is penalised by both p_{20} and p_{21} and free timeslots that directly follow are only penalised by p_{20}. This stimulates joining free periods together: XXOOOXX is preferred over XOXOXOX.

Number of penalties: One for each time a class has a free timeslot preceded by a timeslot with an event it must attend assigned to it. (Not applicable during lunch periods and after the last event for a class on a day.)

$$p_{21} * n_{21} = p_{21} * \sum_{C} \sum_{D} \left(\sum_{t=x}^{\min(lunch(d))-1} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}, 0} * \left(1 - \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t-1} * \delta_{c(e),c}, 0}\right) \right.$$
$$\left. + \sum_{u=1+\mathrm{m}\ (lunch(d))}^{y} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c}, 0} * \left(1 - \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u-1} * \delta_{c(e),c}, 0}\right) \right)$$

$With$:

$$x = \max\left\{(first\ timeslot\ on\ d), 1 + z * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{z} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right), 0}\right)\right\}$$
$$y = \max\left\{z * \left(1 - \delta_{\left(\sum_E \sum_{t=z}^{last\ on\ d} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right), 0}\right)\right\}$$
$$z = \{first\ timeslot\ on\ day\ d, \dots, last\ timeslot\ on\ day\ d\}$$

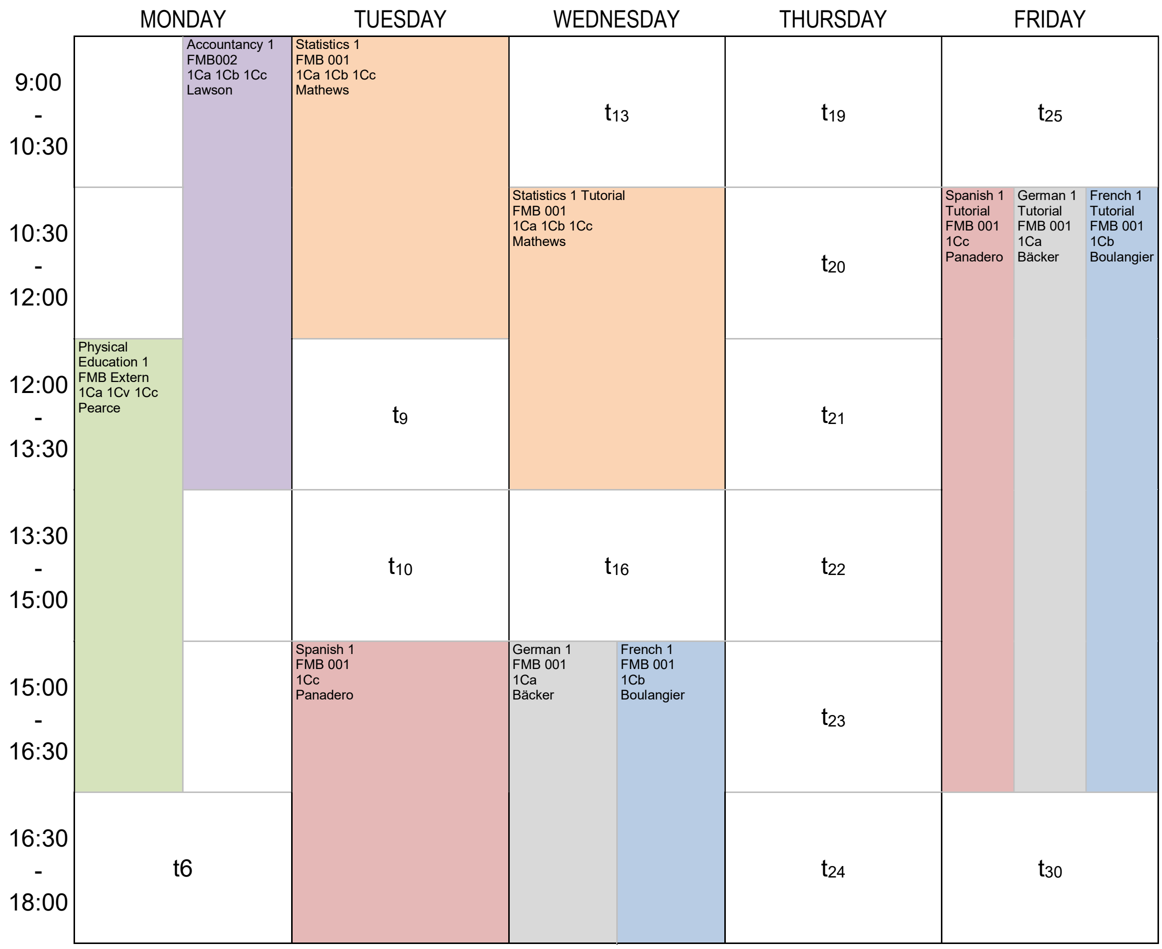
3.3.8 EXAMPLES PENALTY CALCULATION

The example in this paragraph shows how to count the number of penalties in a small problem instance. We will use two representations of a part of the timetable: the timetable for a class and one for a lecturer. The first few penalties will be explained as if the timetable below (of class 1C) contains all events in the timetabling problem.
The timetable below contains all events of class 1C during a week. (Five days with six timeslots each). Each coloured block represents a lecture, with its course name, location, attending groups and lecturer given in the upper left corner. Students are split into sub groups (1Ca, 1Cb, 1Cc) based on their elected language: German, French or Spanish.
Events that share a timeslot may cause a scheduling conflict (during the third timeslot on Monday Accountancy 1 and Physical Education 1 conflict for all groups) but that does not have to be the case (German 1 and French 1 on Wednesday afternoon are attended by different groups.
The following table gives the event, room and class characteristics that are used throughout the example calculations. The preferred lecture hours and lunch times are also given.

e	Type(e)
Accountancy 1	1
French 1	1
French 1 Tutorial	3
German 1	1
German 1 Tutorial	3
Physical Education 1	5
Spanish 1	1
Spanish 1 Tutorial	3
Statistics 1	1
Statistics 1 Tutorial	3

r	Type(r)	Size(r)
FMB 001	1	32
FMB 002	1	28
FMB 003	3	32
FMB Extern	5	50

c	Size(c)
1Ca	31
1Cb	31
1Cc	31
Multiple classes	Max{31,31,31}



E={*Accountancy 1, French 1, …* }
R={*FMB 001, FMB 002, FMB 003, FMB Extern*}
$T=\{t_1, t_2 \ldots t_{30}\}$
D={*Monday, Tuesday, Wednesday, Thursday, Friday*}

$Monday = \{t_1, t_2 \ldots t_6\}$
$Tuesday = \{t_7, t_8 \ldots t_{12}\}$
…

$lunch(Monday) = t_3$
$lunch(Tuesday) = t_9$
…

In this example we want to avoid using the first and last timeslot of each day. The intended start and end timeslots are:
$t_{start}(Monday) = t_2,$ $t_{end}(Monday) = t_5$
$t_{start}(Tuesday) = t_8,$ $t_{end}(Tuesday) = t_{11}$
…

Events scheduled in a room of a different type are given a full penalty, except for events of type 1 that are scheduled in rooms of type 2.

$$\boldsymbol{P} = penalty\ of\ assigning\ an\ event\ of\ type\ te\ to\ a\ room\ of\ type\ tr \quad \begin{bmatrix} p_{1,0} & \cdots & p_{te,0} \\ p_{1,1} & \cdots & p_{te,1} \\ \vdots & \ddots & \vdots \\ p_{1,tr} & \cdots & p_{te,tr} \end{bmatrix}$$

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0.2 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Room related penalties

There are two overbookings: FMB 001 on Wednesday afternoon and room FMB 003 on Friday, violating constraint 1 ten times in total.

$$n_1 = \sum_R \sum_T \max\left(0\,, \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} - 1\right)$$
$$= 2 * (2 - 1) + 0 + 4 * (3 - 1) + 0$$
$$= 2 * 1 + 4 * 2 = 10$$

Constraint 2 is violated three times. Only room FMB 002 is too small to accommodate all students: For e = Accountancy 1, r = FMB 002, t = 1, 2 or 3 holds that $\delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{room\ size(r),i} = 1$, which is zero in all other instances of e, r, t and i.

$$n_2 = \sum_E \sum_R \sum_T \sum_{i=0}^{group\ size(e)-1} \left(\delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{room\ size(r),i}\right)$$
$$= 3$$

Constraint 3 dictates that an event must be assigned to a room of the same type. The only event for which this is not the case is the Statistics 1 Tutorial on Wednesday morning.

$$n_3 = \sum_E \sum_R \boldsymbol{P}(type(e), type(\ \pi_1(\boldsymbol{a}(e)),r)$$
$$= \boldsymbol{P}\left(ty \quad (Statistics\ 1\ Tutorial), type(FMB\ 001) * \delta_{\pi_1(\boldsymbol{a}(e)),FMB\ 001}\right)$$

$$= \boldsymbol{P}(3, (1 * 1)) \quad \rightarrow \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0.2 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix} = 1$$

Constraint 4, all rooms must be used during all timeslots, is violated many times as there are by far not enough events to use all four rooms during all thirty timeslots.

$$n_4 = \sum_R \sum_T \max\left(0\,, 1 - \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t}\right)$$

$$= 6 * (1 - 0) + 2 * (1 - 0) + (1 - 0) + (1 - 0) + 12 * (1 - 0) +$$
$$27 * (1 - 0) +$$
$$25 * (1 - 0) + (1 - 0) +$$
$$2 * (1 - 0) + 25 * (1 - 0)$$

$= 22 + 27 + 26 + 27$
$= 102$

Constraint 5, all seats must be used during each timeslot, is, just like constraint 4, violated a lot. It is important to note that in case of a double booked room, all attending students are combined to see if there are any free seats left. Therefore FMB 001 has two timeslots without any empty seats, six with only one empty seat and of course the 22 timeslots that it is not used, causing 32 empty seats each timeslot.

$$n_5 = \sum_R \sum_T \max\left(0, capacity(r) - \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} * group\ size(e)\right)$$
$= (22 * (32 - 0) + 6 * (32 - 31) + 2 * 0$
$27 * (28 - 0) + 3 * 0$
$26 * (32 - 0) + 4 * 0$
$27 * (50 - 0) + 3 * (50 - 31)$
$= 710 + 756 + 832 + 1407$
$= 3705$

Class related penalties

We will continue with the same timetable and cover the class related constraints, before switching to the timetable of a lecturer.

Constraint 10 prevents scheduling conflicts for classes. In this example the original class 1C split in three classes, based on the language course they take. For each of these classes(1Ca, 1Cb, 1Cc) there is only one conflict: Accountancy 1 and Physical Education 1 during timeslot t_3 therefore the number of penalties is three.

$$n_{10} = \sum_C \sum_T \max\left(0\,, \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\ - 1\right)$$
$= 3 * \big((2 - 1) + 14 * (1 - 1) + 15 * 0\big)$
$= 3 + 0 + 0$

Constraints 12 ensures a free period to lunch, in this case the third timeslot of each day. Which is only true for Tuesday and Thursday and the same for all classes.

$\delta_{\left(\Sigma_E\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0} = 1 \quad for\ t = \{t_9, t_{21}\}$
$1 - \ \delta_{\left(\Sigma_E\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0} = 1 \quad for\ t = \{t_3, t_{15}, t_{27}\}$

$$n_{12} = \sum_C \sum_D \delta_{\sum_{t \in lunch(d)} 1 - \delta_{\left(\Sigma_E\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0},1}$$
$= 3 * 3$
$= 9$

Constraints 13 limits the span of a workday for each class, to e.g. 5 timeslots. That will cause one penalty on Tuesday as class 1Cc start at t_3 and ends at t_{12}, making it a workday of six timeslots.

$$n_{13} = \sum_{C}\sum_{D}\sum_{i=5+1}^{max\,hr\ \ per\,day} \delta_{x,i}$$

$$x = \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$
$$- \max\left\{(first\ timeslot\ on\ d) - 1\ , z * \left(\delta_{\left(\sum_E \sum_{t=fir\ \ on\ d}^{z}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$

$With$:
$y = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$
$z = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

When looking at class 1Cc on Tuesday:
$x = \max\{6, 7, 8, 9, 10, 11, 12\} - \max\{6, 0, 0, 0, 0, 0\}$
$x = 12 - 6 = 6$

In all other instances $x \leq 5$ and therefore $\sum_{i=5+1}^{max\,hr\ \ per\,day} \delta_{x,i} = 0$
$n_{13} = (5 * 0) + (5 * 0) + (0 + 1 + 0 + 0 + 0) = 1$

Constraint 14 ensures a minimum workload whenever a class must attend events during a day. In this example, that minimum workload is set at three timeslots per day. Classes 1Ca and 1Cb only have to attend two timeslots on Tuesday whereas 1Cc attends only 2 timeslots on Wednesday. As only days with one or two timeslots with events are penalised, the free Thursday is not a constraint violation.

$$n_{14} = \sum_{C}\sum_{D}\sum_{i=1}^{min\ timeslots-} \delta_{(6-x),i}$$

$$x = \sum_{t=first\ on\ d}^{last\ on\ d} \delta_{\left(\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right),0} \qquad = number\ of\ free\ timeslots\ on\ a\ day$$

$$(6 - x) = number\ of\ timeslots\ with\ events\ on\ a\ day$$

$n_{14} = 1 + 1 + 1$
$= 3$

Constraint 15 limits the usable timeslots, in this example to the second to the last timeslot of each day.

$$n_{15} = \sum_{C}\sum_{D}(\max\{0, intended\ start\ time\ (d) - actual\ start\ time(d)\}$$
$$+\ \max\{0, actual\ end\ time(d) - intended\ end\ time(d)\})$$

$$n_{15} = \sum_{C}\sum_{D}\left(\max\left\{0, \text{intended}\ start\ time\ (d)\right.\right.$$
$$- \max\left\{first\ timeslot\ on\ d, 1 +\ x * \delta_{\left(\sum_E \sum_{t=fir\ \ on\ d}^{x}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right\}\Big\}$$
$$+\ \max\left\{0, \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}\right.$$
$$\left.\left. -\ intended\ end\ time(d)\right\}\right)$$

$With$:
$x = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$
$y = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

When limited to the Monday and class 1Ca:
$n_{15} = \max\big(0,(2-1)\big) + \max(0,5-6)$
$= \max(0,1) + \max(0,-1) = 1 + 0 = 1$

When limited to the Tuesday and class 1Cb:
$n_{15} = \max\big(0,(8-7)\big) + \max\big(0,(8-12)\big)$
$= \max(0,1) + \max(0,-4) = 1 + 0 = 1$

When limited to the Wednesday and class 1Cc:
$n_{15} = \max\big(0,(14-14)\big) + \max\big(0,(15-18)\big)$
$= \max(0,0) + \max(0,-3) = 0 + 0 = 0$

All classes and days combined gives a total number of six penalties:
$n_{15} = (1+1+0+0+0) + (1+1+0+0+0) + (1+1+0+0+0)$
$= 2+2+2$
$= 6$

Constraint 16 aims at scheduling all events for some classes in the morning. In this example all events attended by classes 1Cb and 1Cc should be in the first three timeslots of each day. The number of penalties is the number of timeslots from the end of the afternoon until the last event of the day for each class in the morning group.

$$n_{16} = \sum_{C \in morning\ group} \sum_{D} \Bigg(\max\Big\{0, \max\Big\{x * \Big(1 - \delta_{\left(\sum_E \sum_{t=x}^{last\ on\ d}\left(\delta_{\pi_2(a(e)),t} * \delta_{c(e),c}\right)\right),0}\Big)\Big\} - third\ timeslot(d)\Big\}\Bigg)$$

$With$:
$x = \{first\ timeslot\ on\ day\ d, \ldots , last\ timeslot\ on\ day\ d\}$

$$n_{16} = \max\big(0,(5-3)\big) + \max\big(0,(8-9)\big) + \max\big(0,(18-15)\big) + 0 + \max\big(0,(29-27)\big) + \max\big(0,(5-3)\big) + \max\big(0,(12-9)\big) + \max\big(0,(15-15)\big) + 0 + \max\big(0,(29-27)\big)$$
$= (2+0+3+0+2) + (2+3+0+0+2)$
$= 14$

Constraint 17 is aimed at concentrating all events for certain classes in the afternoon. Which is in this case class 1Ca, with the last three timeslots of each day taken to be the afternoon. The number of penalties is the number of timeslots from the start of the first event of the day until the start of the afternoon.

$$n_{17} = \sum_{C \in morning\ group} \sum_{D} \Bigg(\max\Bigg\{0, fourth\ timeslot(d) - \max\Bigg((first\ timeslot\ on\ d), 1 + x * \Big(\delta_{\left(\sum_E \sum_{t=fir\quad on\ d}^{x}\left(\delta_{\pi_2(a(e)),t} * \delta_{c(e),c}\right)\right),0}\Big)\Bigg)\Bigg\}\Bigg)$$

$With$:
$x = \{first\ timeslot\ on\ day\ d, \ldots , last\ timeslot\ on\ day\ d\}$

$$
\begin{aligned}
n_{17} &= \max\{0, (4-1)\} + \max\{0, (10-7)\} + \max\{0, (16-(1+13))\} + 0 \\
&\quad + \max\{0, (28-(1+25))\} \\
&= 3+3+2+0+2 \\
&= 10
\end{aligned}
$$

Constraint 18 states that no class can have a day off on either Monday or Friday, all classes comply in this case.

$$
\begin{aligned}
n_{18} &= \sum_{C} 1 - \left(1 - \delta_{\left(\sum_E \sum_{t \in d = monday}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right) \\
&\quad * \left(1 - \delta_{\left(\sum_E \sum_{u \in d = friday}\left(\delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c}\right)\right),0}\right) \\
&= 3 * \left(1 - (1-0) * (1-0)\right) \\
&= 3 * (1 - 1 * 1) \\
&= 0
\end{aligned}
$$

Constraint 19 dictates that the events for a class should be spread over a predefined number of days per week, in this case four days. All three classes comply with this constraint.

$$
\begin{aligned}
n_{19} &= \sum_{C} \left(1 - \delta_{\left(\sum_D \left(1 - \delta_{\left(\sum_E \sum_{t \in d} \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right),0}\right)\right),4}\right) \\
&= 3 * (1-1) \\
&= 0
\end{aligned}
$$

Constraint 20 bans free timeslots between the first and last timeslot with events for a class, excluding lunch times (third timeslot of every day). For class 1Cc, t_{10} causes a penalty and for the other two classes t_{16}.

$$
n_{20} = \sum_{C} \sum_{D} \left(\sum_{t=x+1}^{second\ on\ d} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c},0} + \sum_{u=fou\quad on\ d}^{y-1} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c},0} \right)
$$

$With$:

$$
x = \max\left\{(first\ timeslot\ on\ d),1 + z * \left(\delta_{\left(\sum_E \sum_{t=fir\quad on\ d}^{z}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}
$$

$$
y = \max\left\{z * \left(1 - \delta_{\left(\sum_E \sum_{t=z}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}
$$

$$
z = \{first\ timeslot\ on\ day\ d, \ldots, last\ timeslot\ on\ day\ d\}
$$

$$
\begin{aligned}
n_{20} &= (0+0+1+0+0) + (0+0+1+0+0) + (0+1+0+0+0) \\
&= 3
\end{aligned}
$$

Constraint 21 puts an additional penalty on the first free timeslot in a streak, except when that timeslot is part of a lunch period. Timeslots t_9 and t_{16} are the first timeslots in a free streak, but only t_{16} is not part of a lunch period.

$$n_{21} = \sum_C \sum_D \left(\sum_{t=x}^{second\ on\ d} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c},0} * \left(1 - \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t-1} * \delta_{c(e),c},0}\right) + \sum_{u=fourth\ on\ d}^{y} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c},0} * \left(1 - \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u-1} * \delta_{c(e),c},0}\right) \right)$$

$With$:

$$x = \max\left\{(first\ timeslot\ on\ d)\ ,1 +\ z * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{z}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$

$$y = \max\left\{ z * \left(1 - \delta_{\left(\sum_E \sum_{t=z}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$

$$z = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$$

$$n_{21} = (0+0+1+0+0) + (0+0+1+0+0) + (0+0+0+0+0)$$
$$= 2$$

The last constraint related to classes is constraint 11, which is based on the number of violations of constraints 12 to 21 and therefore saved for last. In this example we assume that those constraint violations cause a penalty p_i of 1 point each and that the maximum penalty per class is 16 points and that every point above that threshold is counted as a violation of constraint 11, n_{11}.

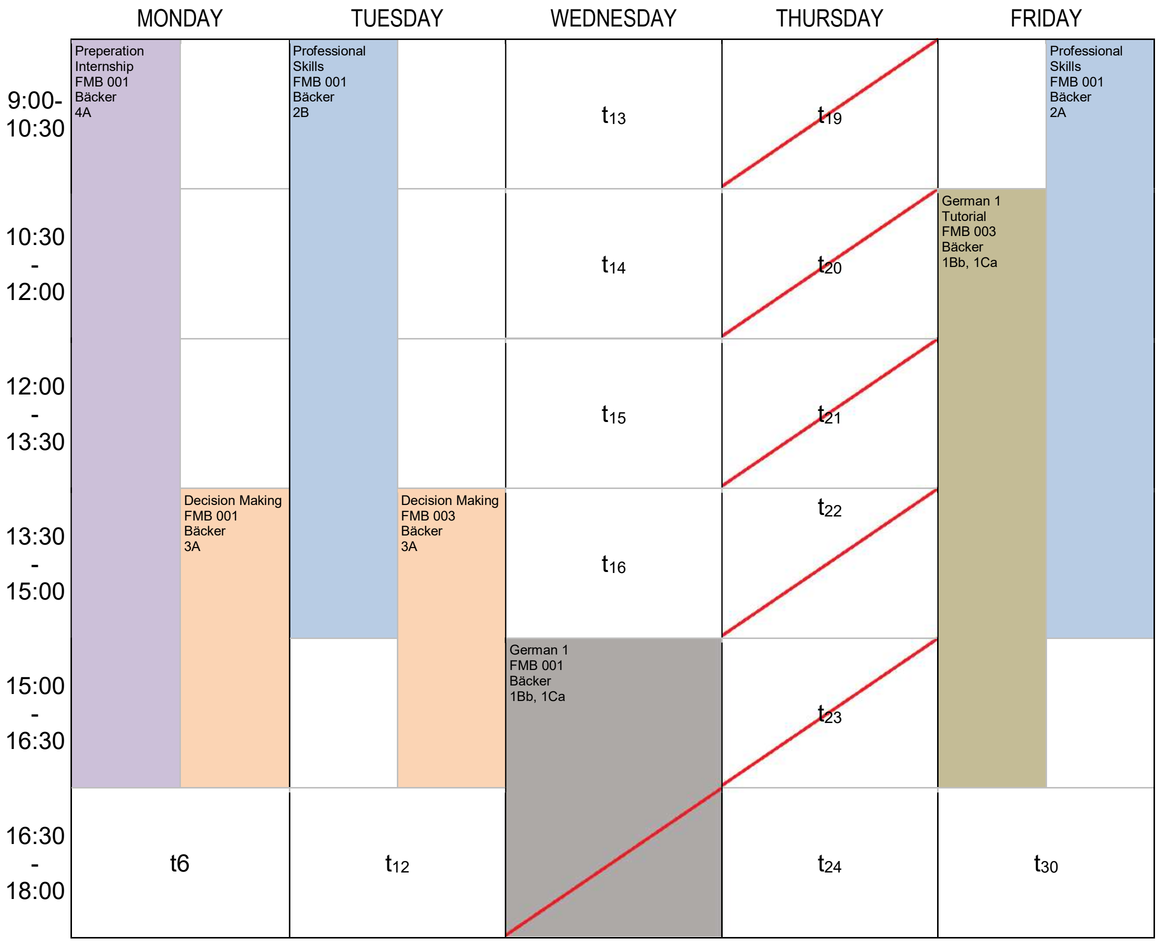
$$n_{11} = \sum_C \max\left(0\ , \sum_{i=12}^{21} \sum_{j=1}^{10} \left(p_i * n_i(c) * \delta_{p_i,j}\right) - 16 \right)$$

$$p_i * n_i(c) * \delta_{p_i,j} = n_i(c) \quad for\ all\ i\ and\ j = 1$$
$$p_i * n_i(c) * \delta_{p_i,j} = 0 \qquad for\ all\ i\ and\ j \neq 1$$

$$n_{11} = \max\{0, (3+0+1+2+0+10+0+0+1+1) - 16\} + \max\{0, (3+0+1+2+7+0+0+0+1+1) - 16\} + \max\{0, (3+1+1+2+7+0+0+0+1+0) - 16\}$$
$$n_{11} = \max\{0,2\} + \max\{0,-1\} + \max\{0,-1\}$$
$$= 2 + 0 + 0$$
$$= 2$$

For the staff related penalties, we will limit ourselves to a single staff member, Mrs. Bäcker, whose timetable is shown below. Each coloured block represents an event she teaches. We have listed the course name, room, lecturer and classes in the upper left corner of each event. The red lines in timeslot t_{18} to t_{23} indicate that Mrs. Bäcker is not available to teach during these timeslots. All things from the previous example, e.g. rooms, lunch times, remain the same in this example. In addition we the set of lecturers has one item in it and the set of events is changed.

L={Bäcker}
E={Preperation Internship, Professional Skills, …}.



Staff related penalties

Constraint 6 limits the number of events a lecturer can teach during a timeslot to one. This limit is exceeded on Monday and Tuesday afternoon and Friday.

$$n_6 = \sum_L \sum_T \max\left(0\,, \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l} \; - 1\right)$$
$$= 2 + 1 + 0 + 0 + 3$$
$$= 6$$

Constraint 7, no one needs to teach when not available, is violated on Wednesday afternoon.

$$\boldsymbol{U}\,(l,t) \;=\; 1 \quad if\ lecturer\ l\ is\ not\ available\ (in\ this\ case\ for\ t = \{t_{18}, \dots, t_{23}\})$$
$$= \; 0 \quad if\ lecturer\ l\ is\ available\ during\ timeslot\ t$$
$$n_7 = \sum_T \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \boldsymbol{U}(l(e),t)$$
$$= 17 * 0 + 1 + 12 * 0$$
$$= 1$$

Constraint 8 ensures a free timeslot to lunch, the third every day. This constraint is violated on Monday, Tuesday and Friday.

$$n_8 = \sum_L \sum_D \delta_{\sum_{t \in lunch(d)}(1-\,\delta_{\left(\sum_E\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}),1}$$

$= 1 + 1 + 0 + 0 + 1$
$= 3$

Constraint 9 limits the span of a workday to a certain number of timeslots, e.g. 4.

$$n_9 = \sum_L \sum_D \sum_{i=4+1}^{max\ hrs\ per\ day} \delta_{x,i}$$

$$x = \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(a(e)),t} * \delta_{l(e),l}\right)\right),0}\right)\right\} - \max\left\{(first\ timeslot\ on\ d) - 1\ , z * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{z}\left(\delta_{\pi_2(a(e)),t} * \delta_{l(e),l}\right)\right),0}\right)\right\}$$

$With:$
$y = \{first\ timeslot\ on\ day\ d, \ldots\ , last\ timeslot\ on\ day\ d\}$
$z = \{first\ timeslot\ on\ day\ d, \ldots\ , last\ timeslot\ on\ day\ d\}$
$n_9 = 1 + 1 + 0 + 0 + 1$
$= 3$

3.3.9 PARAMETER SETTINGS

There is no universal rule for parameter settings of simulated annealing heuristics. To determine the run length we have run the model with a wide variety of run lengths. We have chosen an arbitrary set of rooms from the middle of our area of interest, with U_R = 40.2% and tested each run length under various parameter settings.
Table 3.3-1 gives an overview of the parameters that we examined. Testing all combinations would take 768 runs. We chose to limit ourselves to 32 combinations per run length, adding up to 192 runs. Table 3.3-2 shows which combinations we have examined. We have calculated the average penalty score of all runs that share a parameter setting, this shows which parameters have a large impact on the penalty score and which levels give the best results. The outcomes can be found in Figure 3.3-6.

TABLE 3.3-1 EXAMINED PARAMETER SETTINGS

Parameter	Levels			
t_{start}	10	100	1000	10000
t_{end}	0.01	0.1	0.25	0.33
w_{min}	0.01	0.25		
r_{min}	0.01	0.25		
iterations/temperature	7	70		
$iteration_{max}$	1000000	3000000	5000000	9000000
	11000000	13000000		

TABLE 3.3-2 EXAMINED PARAMETER LEVEL COMBINATIONS

w_{min}	t_{start}	t_{end}	iterat. / t	r_{min}
0.01	10	0.01	70	0.05
0.25	10	0.01	7	0.05
0.01	10	0.1	70	0.005
0.25	10	0.1	7	0.005
0.01	10	0.25	70	0.05
0.25	10	0.25	7	0.05
0.01	10	0.33	70	0.005
0.25	10	0.33	7	0.005
0.01	100	0.01	70	0.05

w_{min}	t_{start}	t_{end}	iterat. / t	r_{min}
0.25	100	0.01	7	0.05
0.01	100	0.1	70	0.005
0.25	100	0.1	7	0.005
0.01	100	0.25	70	0.05
0.25	100	0.25	7	0.05
0.01	100	0.33	70	0.005
0.25	100	0.33	7	0.005
0.01	1000	0.01	7	0.005
0.25	1000	0.01	70	0.005
0.01	1000	0.1	7	0.05
0.25	1000	0.1	70	0.05
0.01	1000	0.25	7	0.005
0.25	1000	0.25	70	0.005
0.01	1000	0.33	7	0.05
0.25	1000	0.33	70	0.05
0.01	10000	0.01	7	0.005
0.25	10000	0.01	70	0.005
0.01	10000	0.1	7	0.05
0.25	10000	0.1	70	0.05
0.01	10000	0.25	7	0.005
0.25	10000	0.25	70	0.005
0.01	10000	0.33	7	0.05
0.25	10000	0.33	70	0.05

Run length

In addition to the effect of each parameter setting, we also looked into the interaction between two parameter settings. The average penalty score of all runs with two specific parameter levels are taken and compared to the other levels of these two parameters. This is visualised in Figure 3.3-7. Each graph shows the interaction of two parameters with one series or point at the horizontal axis per level and the average penalty score on the vertical axis.

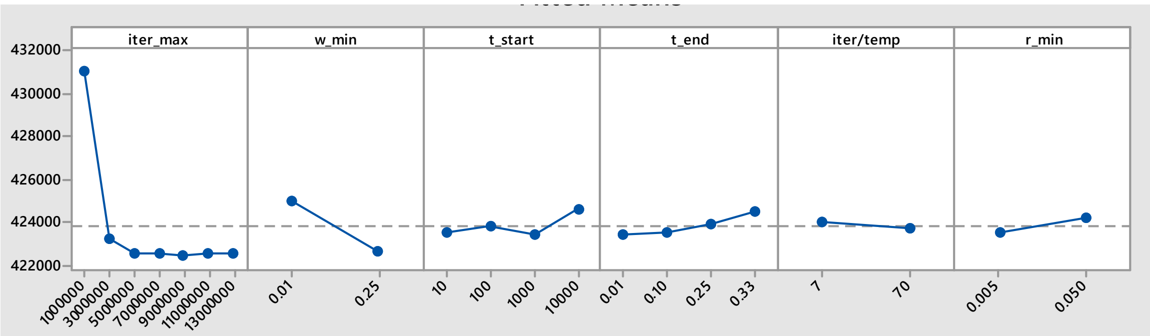


FIGURE 3.3-6 MAIN EFFECT OF PARAMETER SETTINGS ON PENALTY SCORE.

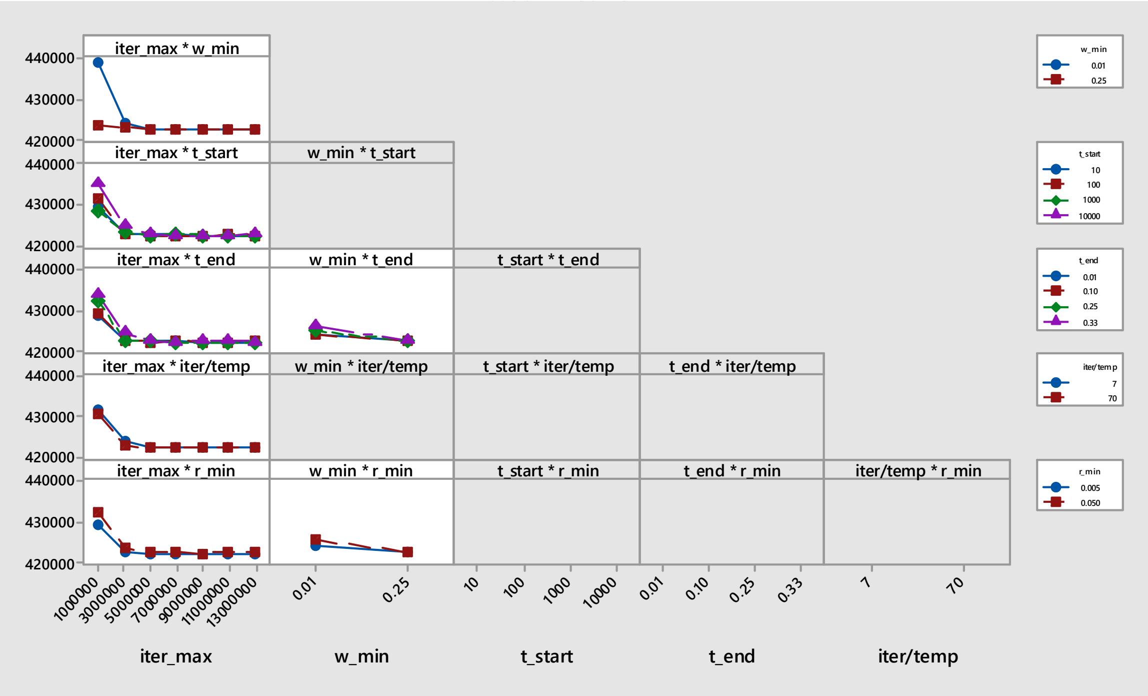


FIGURE 3.3-7 EFFECT OF PARAMETER INTERACTION ON PENALTY SCORE.

A quick glance at Figure 3.3-6 or Figure 3.3-7 shows that the most important parameter setting is the run length, iterationmax. We started with 1,000,000 iterations and increased that number in steps of 2,000,000 until no real improvement was made.

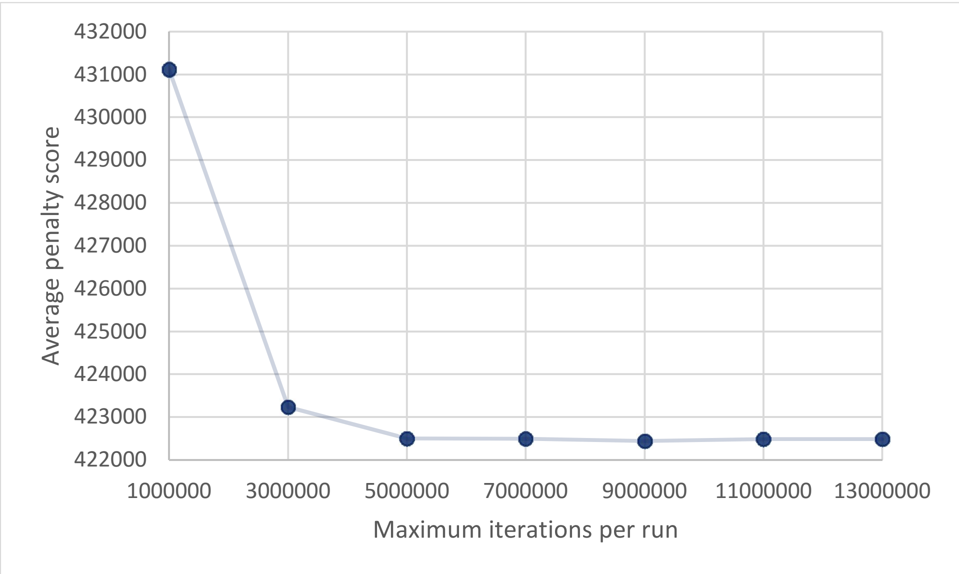


Figure 3.3-8 shows the average penalty scores in more detail. The horizontal lines are spaced 1000 penalty points, the penalty for a high level constraint violation, apart. All of the average penalty scores of run lengths above 5,000,000 are much less than 1,000 points apart. However, we decided to choose a safe iteration$_{max}$ of 8,000,000. The extra iterations give the heuristic some safety margin to cope with more constraints and variation in the search process.

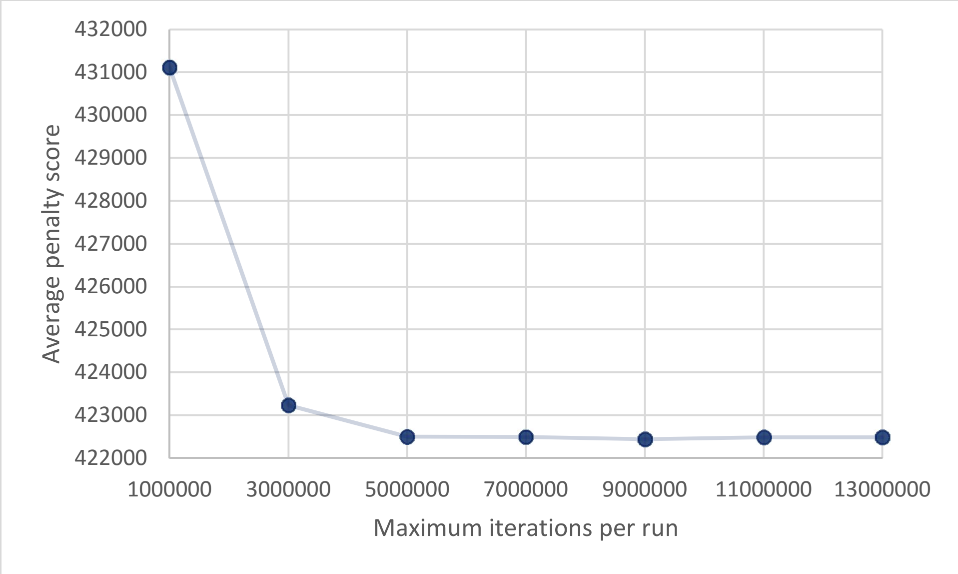


FIGURE 3.3-8 EFFECT OF THE NUMBER OF ITERATIONS PER RUN ON THE PENALTY SCORE

Start and end temperature

The temperatures for simulated annealing always have to be tuned to the problem. We chose the levels of our start and end temperatures by looking at the acceptance probabilities of three penalty scores under various temperatures. Deterioration $\delta = 1000$ would be getting one penalty for breaking a hard constraint, $\delta = 250$ corresponds to not allocating an average class of 25 students to a room for one hour and $\delta = 1$ is a penalty for breaking a soft constraint.

We have chosen the start temperatures such that we include scenarios in which all deteriorations have a high probability to be accepted down to one where only soft constraints violations are accepted. The proposed t_{start} and the corresponding acceptance probabilities can be found in Table 3.3-3.

In the final stage of the annealing process, we do not want to accept any high of mid-level penalties anymore. Therefore we try four t_{end} that have different acceptance probabilities of a 1 point deterioration of the solution value.

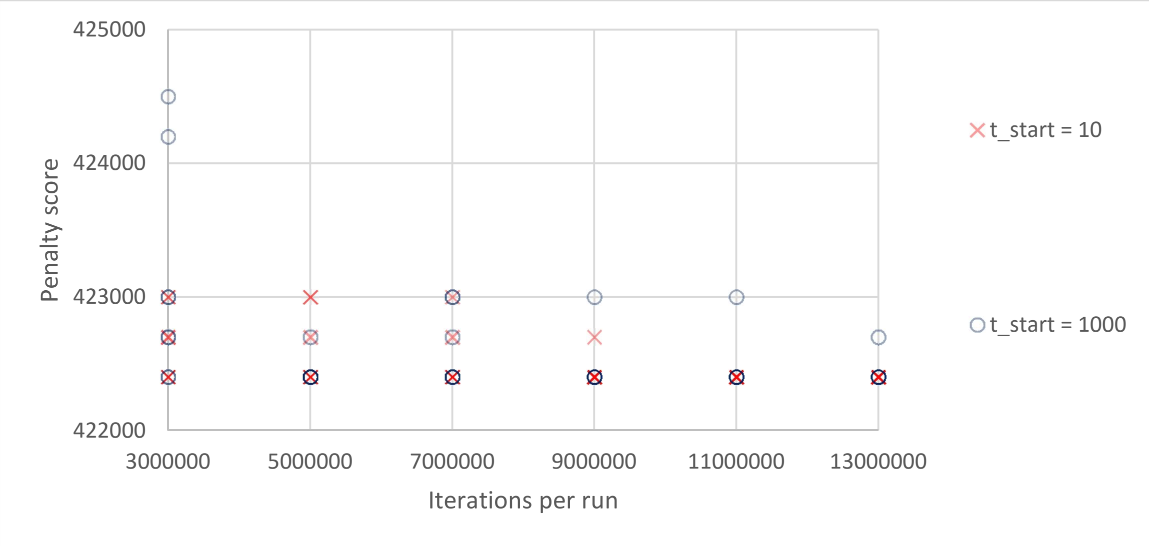
TABLE 3.3-3 TESTED ACCEPTANCE PROBABILITIES AT START AND END TEMPERATURES.

t	δ = 1000	δ = 250	δ = 1
10000	90.48%	97.53%	99.99%
1000	36.79%	77.88%	99.90%
100	0.00%	8.21%	99.00%
10	0.00%	0.00%	90.48%
0.33	0.000%	0.000%	4.830%
0.25	0.000%	0.000%	1.832%
0.1	0.000%	0.000%	0.005%
0.01	0.000%	0.000%	0.000%

Figure 3.3-6 indicates that t_{start} should be either 10 or 1000 and t_{end} should be 0.01. When we look into final solution values of the runs, we see that a lower starting temperature leads

more consistently to a low penalty score. Therefore we choose for a starting temperature of 10.

It is worth noting that the low start and end temperature the simulated annealing heuristic is resembling a hill climbing heuristic. The chances of accepting a candidate solution that adds several new penalties is negligible throughout the run. We choose the best temperature based on our trial runs. It may be wise to run some additional test runs if a new set of experiments with a lot of constraints is done. A high (start) temperature helps to escape local optima which is difficult with our settings.



Other settings

We described in section 3.3.4 that we start reheating if the fraction of accepted solutions per learning period is lower than r_{min}. Bai et al. (2012) found 0.001 to be a suitable value. We want to know if a higher acceptance ratio and therefore more reheating periods is useful therefore we chose to evaluate r_{min} = 0.005 and r_{min} = 0.05.

The iterations per temperatures are recommended to be the number of local operators, which we expect to work fine, out of curiosity we also try ten times more iterations per temperature, giving a more stepped temperature curve: 7 and 70 iterations per temperature.

The various parameter levels we tried had little to no interaction and had only a small influence on the solution value of a run. Therefore we just choose the level with the lowest average penalty scores and come to the following parameter settings:

t_{start}	Start temperature	10
t_{end}	End temperature	0.01
	Iterations per temperature	70
w_0	Minimum weight factor	0.25
r_{min}	Minimum acceptance ratio	0.005
$iteration_{max}$	Iterations per run	8,000,000

3.4 INPUT DATA

The outlines of the model given in section 3.1, can be summarised as scheduling sets of educational events in sets of rooms which aim for different utilisations of the available space. Utilisation is use per capacity so one can vary utilisation by changing either the number of events or the number of rooms. This section explains how we came to our input event and room sets.

3.4.1 VARYING UTILISATION

We limit ourselves to the programmes of FA and IBS located at the Fraijlemaborg, therefore we base the input on historic data of these programmes and the current layout of the Fraijlemaborg. More specifically we use week 37 of 2014. We calculated that the utilisation was 34.7% and the frequency was 42.1% was that week. If we want to assess utilisations between 15% and 75%, the major challenge is to increase the requested utilisation. This can either be done by increasing the usage by adding events (increasing classes and lecturers) or reducing the number of rooms.

We chose to reduce the available space by varying the size of the room set R because of computation time and the interconnectedness of events, classes and lecturers. When we increase the number of events trifold, the problem gets three times as much variables which increases the computation time dramatically. On the other hand, cutting the total room capacity by two thirds will also get a requested utilisation of 75% while keeping the problem size constant. With 112 rooms in use during the academic year 2014-2015, the room pool is large enough to keep a good mix of room sizes and types. However a mismatch between room sizes and types with the event set can have a big impact in the quality of a timetable. The chance of such a mismatch becomes larger with a heavily reduced room sets, this will be addressed in section 3.4.3.

When varying the set of events we must be able to add and remove separate or small groups of events. However, each event is part of multiple groups: a course, the programme of a class, the work schedule of a lecturer. We either get huge groups of interlinked events or we get an imbalance in work load per class or lecturer or incomplete courses. Repairing such imbalances is difficult, especially when we take into account that lecturers only teach a limited number of courses, classes follow a varied curriculum etc. Creating balanced event sets in a large number of different sizes is both difficult and time consuming.

As both options yield the same result, we looked at the drawbacks. The high computation time and finding the right balance in the event sets point us to varying the set of rooms while keeping an eye on the imbalance between room types and sizes. Therefore we used historic data for the set of events and created our own sets of rooms.

3.4.2 SET OF EVENTS

We base our events on historic data. This historic data is an export of the timetabling system of the AUAS for a certain week. Some events are removed from the dataset as they are not part of regular teaching activities. We removed the following events:

- Events without a class assigned it.
- Events without a lecturer.
- Events without a location.
- Events that do not start on normal lecture times.

This includes a small number of incidental events and travel time to external locations, which were separate events in the timetable. The travel times have been included in the model by blocking the timeslot before and after each event at an external location.
Furthermore we removed the duplicates in the set, keeping unique combinations of time, place and event name. In case events have the same place, time and name but different classes or lecturers, we assigned them all to that event. (A Q&A session for everyone who takes Introduction to Accounting becomes one event with multiple classes.)
Removing incidental and incomplete events leaves us with a set of events that represents a typical week of educational events. In practice, a utilisation of 100% will not be achieved and not all spaces in the building are included in the room pool, leaving some wiggle room for incidental room bookings. Bundling multiple classes or lecturers in one event does not change the underlying data.
In the historic timetables, certain events for the same class have be scheduled in the same timeslot (e.g. lectures in French and German). When we ignore this, all courses will be scheduled sequentially, which limits our options, even though there is no need for it. Creating exceptions and checking if an exception applies whenever an assignment of an event is changed requires quite some computational time.
The quickest and most reliable way to create and work with these parallel scheduled courses is by splitting the classes in multiple groups e.g. Class 1Aa takes French and 1Ab takes German in the morning while both 1Aa and 1Ab take Statistics together in the afternoon. This resembles reality of electives where all students must pick one of a few foreign languages. As we do not have data on how classes are split into groups or how big bundled classes are, we assume all subgroups have the same size as the original class and bundled classes to have the same size as the largest class present. (Multiple classes have been scheduled together in a normal size lecture room, which ought to be too small in the historic timetables.)
In the experiments we will sometimes limit ourselves to the programme Business Economics. In these cases we will use a subset of the events and exclude all events that are not attended by classes from the BE programme.

3.4.3 SET OF ROOMS

We took all rooms from the timetable export and added the ones that were not used in that period but are listed as lecture rooms. Each room has a size and a set of features like audio equipment, computers for the students or a location outside the Fraijlemaborg. The room sizes can be used as is but the room types have to be chosen. We distinguished five types based on the qualifications of the timetabling department and the suitabilities listed for each room. These types break the rooms into large groups that are really different and unsuitable for other types of events. We expect that timetablers can manage to schedule events in the best room within each type. This can be refined later, if deemed necessary.

1 - Basic lecture room

  This includes all rooms qualified as a “collegezaal” or “theorielokaal” and may or may not contain a beamer, smart board, audio equipment.

2 - Computer room

This includes each "Computerlokaal" and is equipped with a computer for each student. This type can be excluded in the future if all students will use their own laptop computer.

3 - Specialty room

This includes all rooms qualified as either "Onderwijsruimten domeinspecifiek" or "Pract/Talen" which are in the Fraijlemaborg rooms with audio booths for language courses.

4 - Project room

These are small rooms for (project) meetings or other group work with a capacity of six people without any equipment.

5 - External Location

This are all sports facilities which are not located in the Fraijlemaborg. These will be used in penalty scores so events of this types are included in the timetable but will be excluded when calculating utilisations. Events scheduled here must be preceded and followed by a free timeslot to allow for travel time they are not located in the Fraijlemaborg.

We want various sets of rooms such that we have an even spread of requested utilisation and requested frequency between them. In order to do that we chose an interval of F_R that we want to cover with the sets. We can calculate the number of timeslots needed for all events in our set and with the equation below we can calculate how many roomslots we need.

$$Requested\ frequency, \qquad F_R = \frac{roomslots\ needed\ for\ set\ of\ events}{total\ roomslots\ available}$$

A short example: We have 20 events, each with a duration of 2 timeslots and our week consists of 10 timeslots. This means that the number of rooms in the set must be:

$$rooms\ needed = \frac{\left(\frac{roomslots\ needed\ for\ set\ of\ events}{F_R}\right)}{timeslots\ per\ week} = \frac{\left(\frac{20 * 2}{F_R}\right)}{10} = \frac{4}{F_R}$$

Let us assume that we want to examine requested frequencies on the interval 20% to 60% and that we will use five sets of rooms. In that case we will need one set of 20 rooms, one of 13.333 rooms, one of 10, one of 8 and one set of 6.667 rooms. As one cannot use partial rooms, we round the number of rooms to the nearest integer.

With the number of rooms known, we need to determine their sizes and types. We determine the number of rooms per type by multiplying the total number of rooms in the set with the ratios of event types in the set of events. E.g. if half of the events is of type A, than half of the rooms should be of type A. Fractions are neglected in this stage. After picking the integer number of rooms for each type, we are left with remainders, which add up to an integer number. We try to add a room to each of the types and add it to the type for which the maximum difference of the frequency for each type with the requested frequency is minimum.

Consider the following example: We have an original room set consisting of 4 rooms of type A, 2 rooms B and a room C, 7 rooms in total. There are 10 timeslots in a week and our set of event consists out of 30 events A, 12 events B and 8 events of type C.

Our goal is to create a set of rooms with a requested frequency of 80% consisting of 6 rooms, a factor 0.857 smaller. This means that our new set should include 3.42 rooms A, 1.71 rooms B and 0.86 room C.
First we pick 3 rooms A and 1 room B, giving a total of 4 rooms. Then we add a room C because otherwise none of the events of type C can be scheduled.
This leaves us with 1 room to fill. We try adding a room of each type and minimise the maximum deviation from F_R over all the types. The table below shows that adding a another room of type B is the best option with a maximum deviation of 20%.

Type	Rooms	F_R	$\lvert F_R$-80% $\rvert$	Rooms	F_R	$\lvert F_R$-80% $\rvert$	Rooms	F_R	$\lvert F_R$-80% $\rvert$
A	4	75%	5%	3	100%	20%	3	100%	0%
B	1	120%	40%	2	60%	20%	1	120%	40%
C	1	80%	0%	1	80%	0%	2	40%	40%

The room sets should have similar characteristics as the Fraijlemaborg, therefore only rooms of existing size and type combinations will be used, preferably in the same ratios. The sizes can be used to alter the U_R of the event set and room set combination. We calculate a target U_R for each room set using the average room size. If we have to choose a room size, we pick a room that will move U_R towards its target value, e.g. when all rooms so far are above average and U_R is smaller than its target, pick a smaller room to increase the value of U_R.

$$U_R = \frac{seat\text{-}hours\ needed\ for\ set\ of\ events}{total\ seat\text{-}hours\ available} = \frac{seat\text{-}hours\ needed\ for\ set\ of\ events}{rooms * average\ size * timeslots}$$

Per type we do the following: First we determine the multiplication factor needed to get from the original set of rooms (e.g. the Fraijlemaborg) to the newly created set of rooms and multiply all the numbers of rooms of each size of the current type with that factor. The truncated outcomes are added to the new set of rooms. The remainders add up to an integer number: the number of rooms still to be assigned a size. If there are still more than one room waiting for a size, assume all but one to have the average size of rooms of the current type. The last room will be assigned a size. We will try all possible sizes and calculate a U_R in each case and assign the size that led to the utilisation closest to the target value of U_R.

The steps described above have been used to create a large number of room sets (100 for the events of all programmes, 48 when we only examine the Business Economics programme). Requested utilisation and frequency will increase linearly, which will help to give a clear picture in which we can find the critical utilisation or frequency.

3.4.4 INITIAL SOLUTION

Simulated Annealing needs an initial solution which it will improve to a near optimal solution. Ideally, every experiment gets a unique start solution tailored to its exact needs (penalty structure). However due to the large number of experiments we limited ourselves to two types of initial solution. The first is used under normal circumstances while the second is used when classes are split in a morning and afternoon group.

In both cases we randomise the order of the events to avoid a bias in favour of certain classes, programmes or staff members. The rooms are sorted on size and type to enable us to assign the best suited events to them.
We start at the top of the randomised list of events and look for usable timeslots. An event will only be assigned to a timeslot if the lecturer is available during that timeslot and the class, and lecturers have no other events to attend at the same time. We start with the sixth timeslot on Monday and if that timeslot can be used, we try to find a room that is not occupied during the timeslot, is large enough to accommodate all attendees and is of the same type as the event. If such a room is found, the event is assigned to the room and timeslot, if not we try again with the next timeslot.
Our second try is the sixth timeslot on Tuesday, then Wednesday, Thursday and Friday. If necessary, we will repeat with the seventh timeslot of each day, then the fifth, eight, fourth etc. I.e. we work our way out from the middle of the day. Which is a student preference but also helps to comply with constraint 15: all event should be scheduled between e.g. timeslot 3 and 10 of each day.
In case an event remains unassigned after trying all rooms, it will start as an unallocated event.
The second type of initial solution works roughly the same. The only difference is that the timeslots do not alternate (6,7,5,8,4 etc.) but decreases or increases from timeslot 7 depending on whether the event is attended by classes in the morning or afternoon group (7 to 1 or 7 to 12 respectively).

3.5 SUMMARY OF THE APPLICATION OF QUANTITATIVE MODELS

We use the ideas of Beyrouthy et al. (2006), as explained in section 3.1, to predict how much space is needed to schedule a given set of events. In order to do so we need to schedule as much of those events in a timetable using a varying number of rooms. By doing so with different scheduling rules we can compare the impact of these rules on the minimum required space to facilitate all events.
We have automated the timetabling due to size of the scheduling problem and the large number of instances that must be solved. For that purpose we devised a penalty structure that awards incorporating events in the timetable and penalises violations of a set of rules. There are numerous heuristics to solve minimisation problems, like the minimisation of the penalties we proposed. Section 3.2 describes how we came to our preferred heuristic, Simulated Annealing.
The penalty structure described in section 3.3.7 covers all aspects of timetabling from restricting policies (no double bookings, staff availability) to user preferences (no free periods). By using a penalty structure that favours complying to hard constraints over including an event in a timetable and soft constraints that will not cause rejection of events from the timetable, we can create timetables that include various policies and preferences. This answers the research question:

IV. *How can user preferences and restricting policies be combined into timetables?*

4 RESOURCE BOTTLENECKS

In this chapter we check if we can find a clear bottleneck in the educational logistics. Section 4.1 explains which bottlenecks are perceived throughout the organisation. Sections 4.2 and 4.3 address the impact of these bottlenecks.

4.1 CLAIMS

There are three main claims: the quality of the timetables is too low (Executive Board, 2014c), the available workforce makes it hard(ly possible) to make a timetable (according to the timetabling department) and the operations bureau creates its own problems by reducing the facilities capacity too far (according to the educational branch).

It is remarkable that the requirements of a good timetable could not be presented by either the scheduling department or by the educational branch. It is virtually impossible to actively and effectively pursue better timetables as long as there is no consensus on what makes a good timetable. Therefore we must make explicit what timetables we consider to be of low and high quality and evaluate timetables with those criteria. This is done in chapter 5.

The other two claims state which of the two limited resources are causing timetabling problems: the number of available lecturers or the number of rooms. The timetabling team mentions the low availability of teaching staff as the cause of their scheduling difficulties, without reflecting on the number of rooms available to them whereas the programme manager and year coordinators claim that the teaching staff availability is maximised and that problems therefore are caused by the limited number of lecture rooms. These different perceptions of the cause of timetabling problems fit in the picture of "us and them" thinking of both the operations and educational branch of the school as mentioned in earlier (Executive Board, 2013c), in both cases "they" have not secured sufficient capacity. We will use the model presented in chapter 3 to check the influence of staffing and housing choices on the critical utilisation.

4.2 ACCOMMODATION

Things that can be varied are the number of rooms, the type of rooms and in theory the sizes of rooms. In the experiments we have run, we varied the number of rooms, during which the proportion of sizes and types is kept constant. To simulate the situation with perfect conditions, we stop penalising certain aspects. E.g. when we have enough rooms of any type during any timeslot we will never award a penalty for a mismatch between event and room type.

Figure 4.2-1 compares three situations:

In experiment *a*, we vary the number of rooms but when an event is scheduled, the room types and sizes will be disregarded. The only exceptions are sport events on external locations, which are penalised when they are scheduled in the Fraijlemaborg, just like regular events that are scheduled on external locations. Furthermore multiple events in the timeslots are not allowed for either rooms, classes or lecturers.

In experiment *b*, events may not have more attendees than the room has seats and must be of the same type as the room. Scheduling conflicts for rooms, classes and staff are also not allowed.

However, in experiment *c* scheduling conflicts of staff and classes are disregarded. We only look at double booked rooms, room types and room size.

These situations are quite similar. With few constraints (only no overlapping events) the impact of these constraints is negligible as experiment *b* and *c* have very similar results. Disregarding room types and size improves the critical frequency. This indicates that it can be worthwhile to tailor the room sizes and types to the demand (the events, which are based on the curricula).

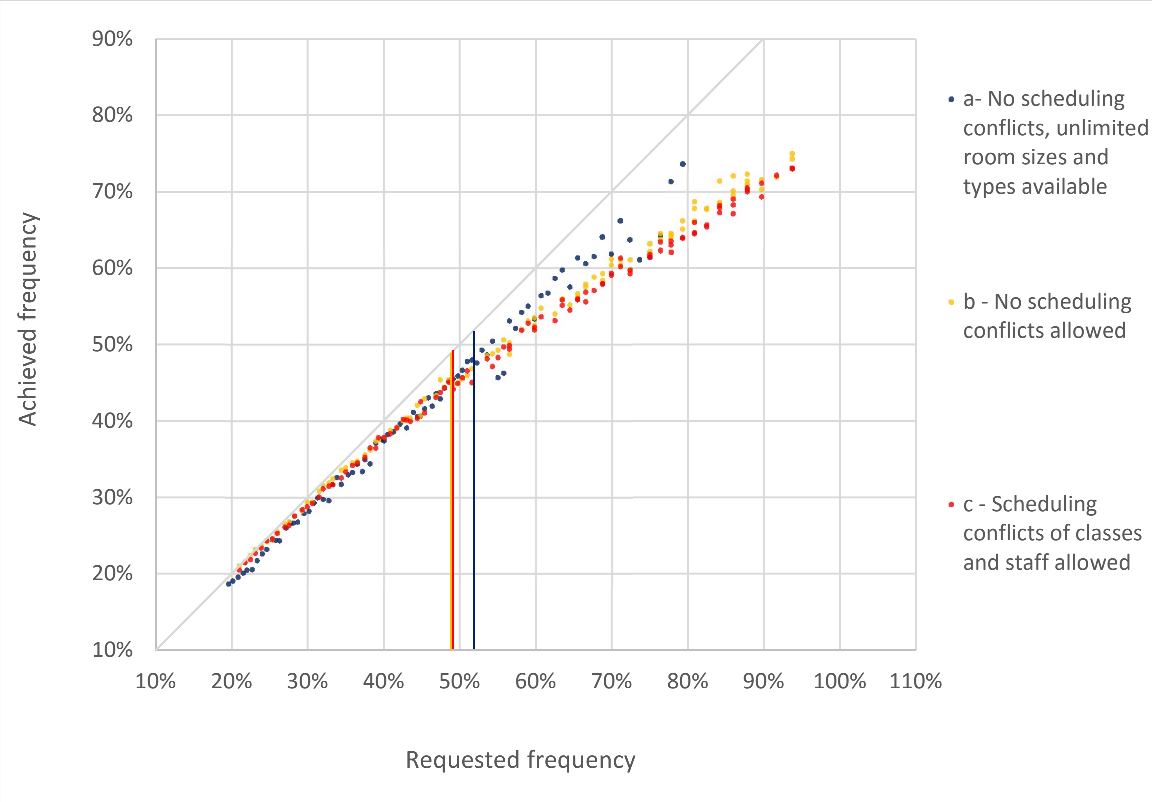


FIGURE 4.2-1 COMPARISON OF CRITICAL FREQUENCIES WITH VARYING ROOM PROPERTIES.

4.3 STAFF

There are some rules that do not apply to rooms but do apply to staff members: labour regulations. The ones that we want to comply with are maximum working day of ten timeslots and a free timeslot to lunch among the fifth, sixth and seventh timeslot of each day. Another rule describes that staff members who work late in the evening (for the part-time programmes) cannot be scheduled before the fourth timeslot of the next day. However, we chose to disregard this principle as we did not include the part-time programmes in this project and it would apply only to a small number of lecturers.

We compared three situations: *s* in which only rooms and classes may not be double booked but lecturers may, *a* in which rooms, classes and staff may not have any scheduling conflicts and *d* in which no scheduling conflicts may occur, staff members have a lunch break, a workday of at most 10 timeslots and are sometimes unavailable.

Experiment s simulates an infinite supply of lecturers, which performs similar to experiment *a*. Experiment *d* performs slightly worse than the other two and starts to deteriorate at a lower frequency. Either the labour regulations and staff availability lower the critical frequency or they make it much more difficult to find a feasible timetable even though they do exist.
Experiment d performs worse at requested frequencies 40% to 48% but has performs better than the other two near 80%. This gives the impression that feasible timetables can be found at high frequencies but very few examples exist. Spreading availabilities and working hours of staff members evenly over the days seems crucial to creating timetables with a high requested utilisation.

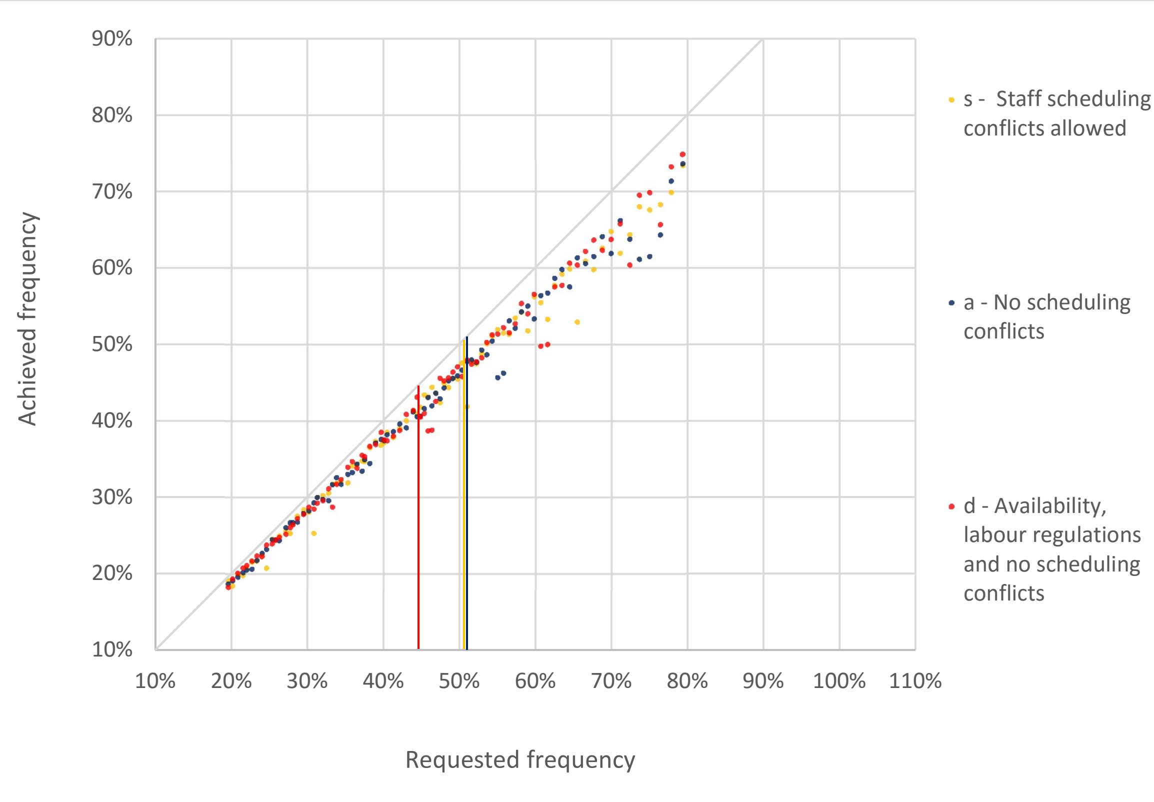


FIGURE 4.3-1 COMPARISON OF CRITICAL FREQUENCIES WITH VARYING STAFF RELATED CONSTRAINTS.

4.4 ACTUAL BOTTLENECK

We have looked at both main resources, staff and space. When the constraints are limited to either staff or space, increasing the number of constraints does not have a much influence. Although labour regulations and staff availability make the achieved frequency quite inconsistent. If we compare the frequency achieved in the actual timetable from 2014/2015 and compare it to the critical frequency of we found (42% versus around 50%), then we are not dealing with under capacity of space. On the other hand, scheduling with the staff availability as it was in 2014/2015 has a negative impact on achieved frequency. Even though some good timetables can be produced at a high requested frequency.

Including staff availability does make it harder to schedule, it is not necessarily an insurmountable obstacle.

This chapter is aimed at one of the research questions:

III. *What are current bottlenecks in the availability of resources?*

Both main resources seem to have sufficient capacity. The exact critical frequency F_C can be debated but with our interpretation of the results, the frequency in the historic timetable of week 37 of 2014/2015, 42%, stayed below the F_C when adding various room related constraints or staff related constraints. Combining numerous constraints and unbalanced supply and demand can cause difficulties for the timetablers. Balancing staff availability and distinguishing staff unavailability from their preference to be free during some timeslots can help avoiding timetabling problems.

5 COMBINING PREFERENCES AND RESTRICTING POLICIES

We have run several experiments and discuss the outcomes in this chapter. A different set of penalties is used in each experiment, Appendix 6 gives an overview of all penalties and input values for constraints are used in each experiment.
As we focus on the Business Economics programme, we limited ourselves to the classes of BE. If the events of all FA programmes are included, it will be mentioned explicitly.

5.1 SCHEDULING ALL PROGRAMMES VERSUS JUST BUSINESS ECONOMICS

As just mentioned, we want to limit ourselves to the Business Economics programme. Which means that the number of events is reduced from 1217 to 317. Both problems have the same penalty structure, mix of events etc. but differ in size. That difference in size may distort the quality of the solutions: The combination of partial solutions may be worse than a solution found for the entire problem.
Figure 5.1-1 shows an example of a problem that can be split in three identical partial problems, each consisting of two short events and a long event. Scheduling the sub problems in the minimum number of rooms leads to a demand for three times two rooms. If we schedule all nine events in the minimum number of rooms we only need four rooms opposed to the six needed for the sub problems.

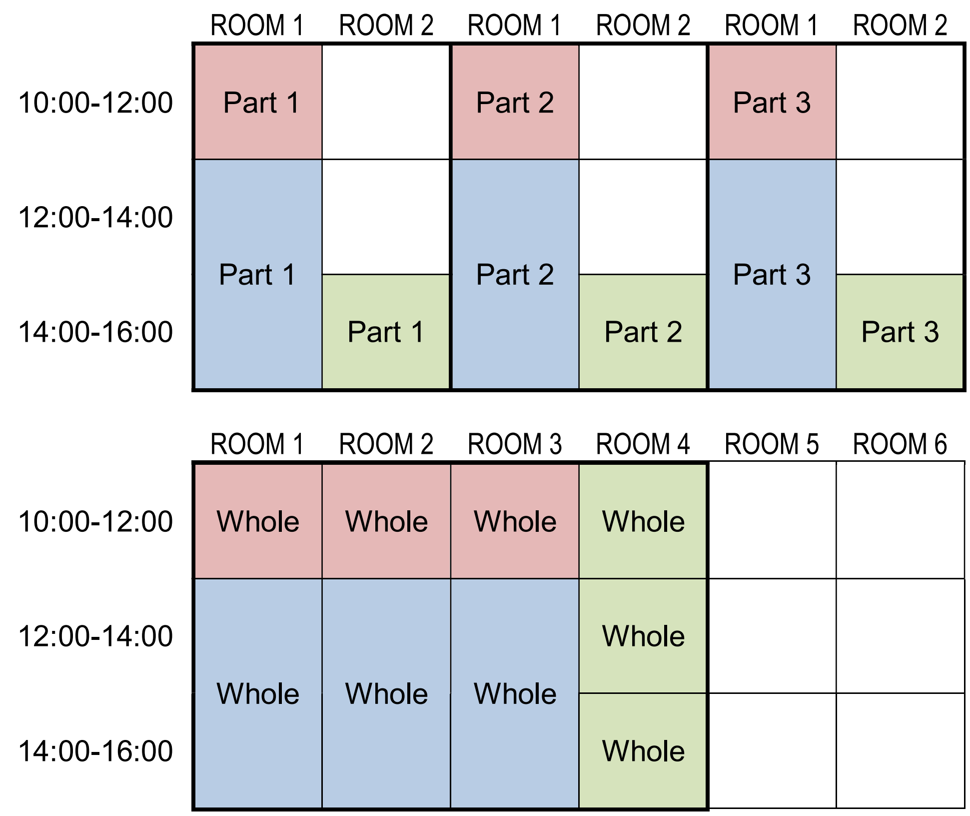


FIGURE 5.1-1 SOLVING PARTIAL PROBLEMS MAY YIELD SIGNIFICANTLY WORSE SOLUTIONS THAN ADRESSING THE PROBLEM AS A WHOLE.

A larger problem instance gives more flexibility, therefore we have run the model under the same constraints with both the events of all programmes of FA and IBS and only the events of BE. The figures below show that the results are very similar.
At high requested frequencies, the sub problem performs slightly worse than the problem instance that includes all programmes. However we find the difference negligible. When we only look at Business Economics we will underestimate the critical frequency. Therefore the decisions based on the results may cause some overcapacity but will not cause space shortage.

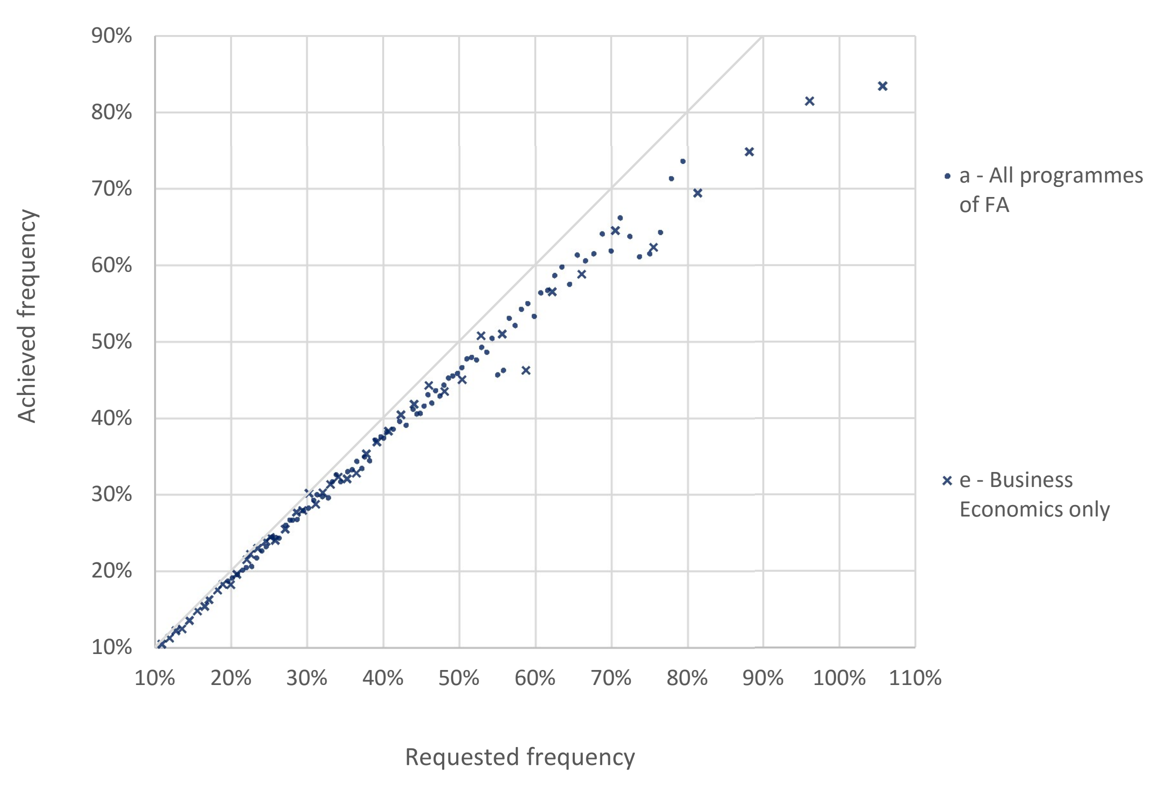


FIGURE 5.1-2 COMPARISON OF A SMALL AND LARGE SCHEDULING PROBLEM UNDER SIMILAR CONSTRAINTS (NO CLASS LECTURER OR ROOM IS SCHEDULED MORE THAN ONCE IN A TIMESLOT)

The comparisons indicate that we can schedule programmes like Business Economics separately from the other events. Which means that we can schedule blocks of events and combine them in a later stadium. This enables timetablers to reserve a fraction of the rooms for each programme and schedule them at different points of time or schedule programmes automatically under different constraints.
It also

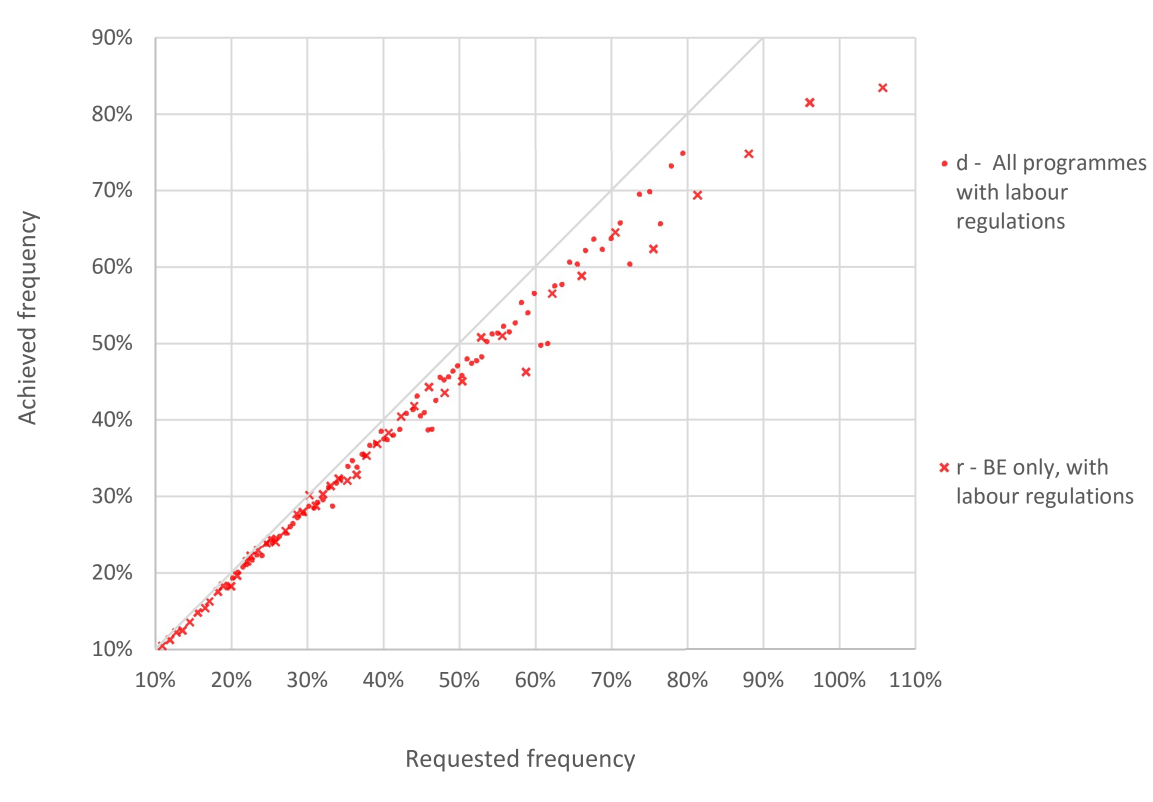


FIGURE 5.1-3 COMPARISON OF A SMALL AND LARGE SCHEDULING PROBLEM UNDER SIMILAR CONSTRAINTS (NO DOUBLE BOOKINGS, AND LABOUR REGULATIONS)

5.2 LIMITED TIME WINDOW

Lecture times, the 60 timeslots per week, are set AUAS wide. The lectures of a class will take between 10 and 30 timeslots per week which may be scheduled during any of the mentioned 60 timeslots. However it may be beneficial to limit that time window to 50 or even less timeslots.

5.2.1 CONCEPT

Both students and staff mentioned the general lecture times as possible improvement. The AUAS uses twelve timeslots between 8:30 and 18:40 for all full-time programmes. With travel times of one and a half hour, which is not exceptional according to the students, students must reserve 7:00 to 20:00 to be sure that they can attend all educational events. Only a small number of timeslots is used in the timetable but work or sports competitions may have already been scheduled before the timetables are published. Using a smaller portion of the week to schedule all events for each class and giving clarity on which students are assigned to which portions far in advance will help students balance their education with their private life.

That balance is also the underlying reason lecturers proposed a smaller time window to schedule all events. Education is part of a student's life, not its whole life and skipping class is an often used solution to scheduling conflicts with private affairs. Improving attendance will help increase study success and speed. A (partial) solution is using lecture times that students prefer.

Just scrapping the first and last timeslot each day results in a 16.6% loss of seat-hours. Which is a significant loss. To prevent that loss of resource capacity but still limit the time window during which a student may have to attend events, morning and afternoon classes are introduced. A morning class has no lectures after 13:40 whereas afternoon classes have no events before 12:50. This halves the time window for each class, without any capacity loss or increased peak demand.

5.2.2 RESULTS

Figure 5.2-1 shows the requested and achieved frequency under different circumstances. We limit ourselves to various sets of timeslots and prevent double bookings, penalty structure can be found in Appendix 6.

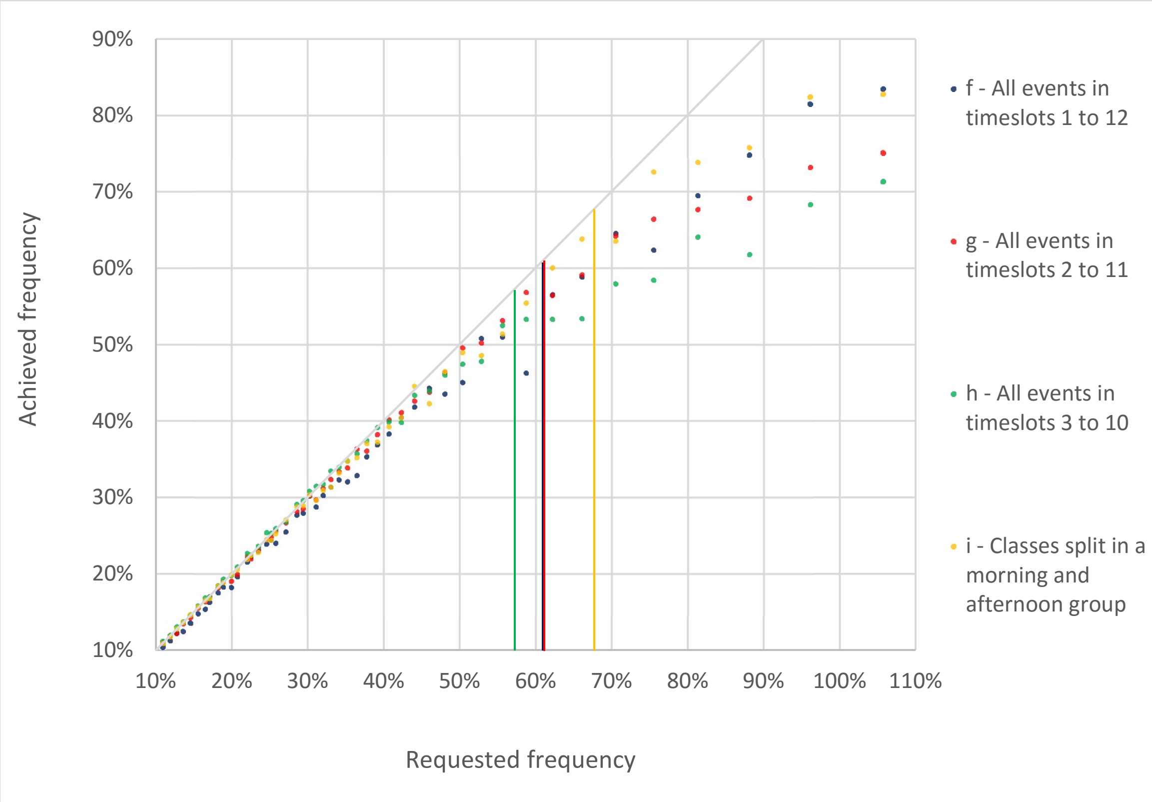


FIGURE 5.2-1 COMPARISON OF CRITICAL FREQUENCIES WHILE USING A VARYING NUMBER OF TIMESLOTS.

When the classes were classified as either a morning or afternoon class, we chose to create equal mixes: one first year class accountancy in the morning, one in the afternoon etc. Lecturer availability had little impact but it may have a high impact if morning classes may be taught by lecturers that only work afternoons. In practice, one would consider this when assigning lecturers to classes. However, as the penalty for lecturer availability had little influence, we chose to exclude it from further experiments instead of going through all class and staff members and pair them again.

At the moment, the timetablers manage quite well to avoid the first and last two timeslots of each day, which corresponds to experiment *g*. One can argue within what margin of the requested frequency the achieved frequency must be before the critical frequency is reached. However experiment *f* and *g* perform quite similar whereas *h* is more clearly worse.

Reducing each day with two timeslots seems achievable (both theoretical and already in practice) but more may cause problems. Splitting the classes in multiple groups e.g.

morning and afternoon classes, helps to achieve a significant higher frequency (and utilisation) than limiting the total number of timeslots.
On a side note, supplying students with timetables far in advance (e.g. six weeks), offering courses multiple times per week and let students enrol in one of the courses may yield similar results. Students can attend lectures at times that suit them without sacrificing roomslots.

5.3 MINIMUM WORKLOAD

One of the reasons for skipping class students mentioned was that they had to travel too long for a single lecture. We tested what would happen if we introduced a minimum number of timeslots. Students have the day off or are offered at least the minimum number of timeslots with events.

5.3.1 CONCEPT

Sometimes travel time exceeds the duration of a lecture. This may be a reason to skip class, which leads to lower attendance, which is undesirable. During the interviews four lecture hours was mentioned as the bare minimum to make a day worthwhile to attend. Therefore we introduce a penalty for every day that a class must attend between 1 and 3 timeslots.

5.3.2 RESULTS

Figure 5.3-1 shows that introducing a hard penalty on a minimum workload has a dramatic impact on the critical frequency. When using morning groups (experiment i and k) the effect is far worse than under normal circumstances (f and j). Possible cause are the fact that an afternoon class must fill four timeslots in a day of only six timeslot, which is much harder than using four out of twelve timeslots.
Furthermore it is worth noting that the value of four timeslots cause serious trouble for some classes. If a class has only four lectures per week of two or three timeslots each, events must be split over one or two days. This means that single events on a day are likely to be unallocated whereas it unlikely that an unallocated event is accepted unless it happens to be scheduled on a day with at least one other event. The latter becomes difficult as there are only three other events which are on the same day (only 1 out of 5 chance) or likely to be also unallocated.
It may be beneficial to apply the minimum workload principle only to selected classes (with many events). Changing from morning and afternoon classes to Monday-to-Wednesday and Wednesday-to-Friday-classes may also help.

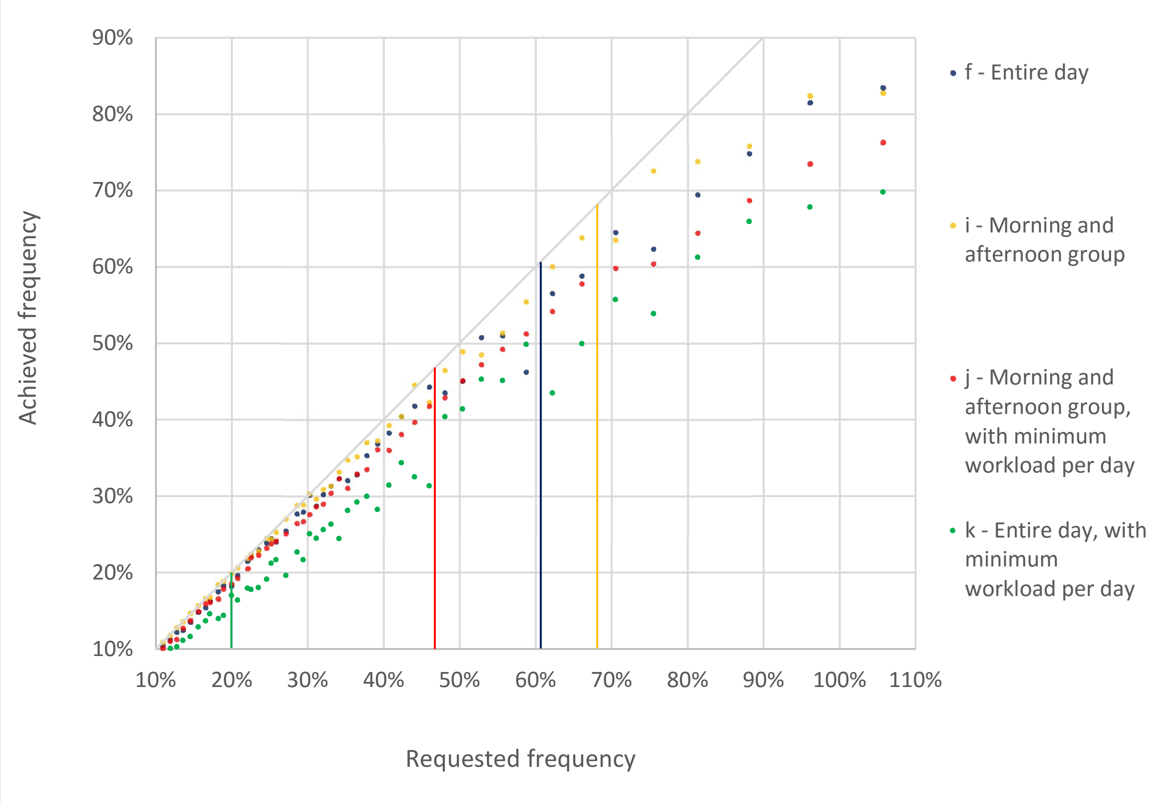


FIGURE 5.3-1 COMPARISON OF CRITICAL FREQUENCIES AFTER INTRODUCING A MINIMUM WORKLOAD PER DAY WHEN A CLASS IS TO ATTEND ANY EVENTS.

5.4 MISCELLANEOUS PREFERENCES

The preferences of students and staff members as mentioned in section 2.6 include a number of measures to increase attentiveness and time on task. With time on task is meant the time a student is working on his courses during a week, both during events and in their own time whereas attentiveness is paying attention and making the most of the attendance of educational events. High attendance, attentiveness and time on task are essential for study success and speed.

5.4.1 SOFT CONSTRAINTS

We included preferences on when events are scheduled in the penalty structure and omitted preferences regarding the content of courses. Classifying all events based on their content is a laborious task which requires intimate knowledge of the content of all courses. We feel that we should concentrate on the position of all events per class in the timetable first. If there results of this project are useful, we can refine the results based on course content afterward. The model can be easily expanded with additional event types. By copying a row in the room-event type penalty matrix we can split events in e.g. taught computer tutorials and independent work computer tutorials. These would have the same characteristics but could be distinguished by a penalty rule that encourages mixing theoretical and practical events on a day (or hard and easy events or boring and exciting events).

The following preferences were used as a penalised constraint:

p_{12} Each class has a lunch break.

p_{13} The workday of a student does not span more than eight hours.

p_{14} A student has either no or at least 4 timeslots with an event each day.
p_{18} All classes have at least one event on Monday and one on Friday
p_{19} The events of a class a spread over four days.
p_{20} A class has no free timeslots between events.
p_{21} A class has minimum alternations between events and free timeslots.

These constraints are given a soft penalty, in this case between one and four. Furthermore we award per class a hard penalty when the total penalty score caused by these soft constraints exceeds a certain number. (See constraint 11 in section 3.3.7)

5.4.2 RESULTS

In experiment *f* all soft constraints are disregarded, in *l* all constraints are given a hard penalty and in experiment *o* a relatively high total score per class, 20, is allowed before the soft constraints are given a hard penalty (exceedance multiplied with 1000). These experiments have similar results, as shown in Figure 5.4-1. However, there is a difference in the average number of constraint violations. Figure 5.4-2 shows that awarding penalties reduces constraint violations but when looking at both figures, one can see that allowing quite a few violations before applying large penalties can drive down soft constraint violations (from 8 to 6 per class) without a negative impact on the achieved frequency (*f* versus *o*).

Lowering the threshold from 20 penalty points per class in experiment *o* to 10 penalty points per class in experiment *m* helps to reduce the number of constraint violations from around five to six to around three to four per class. However, it does also cause great scheduling problems, even under a very low requested frequency.

Experiment *l* has an odd outcome. The achieved frequency is relatively high but the average constraint violation per class lies between three and seven where near zero was expected. We expect that there are a lot of constraint violations in the initial solution. Multiple violations may be caused by an event and swapping it with another event may reduce the total number of violations by one. The swap would be accepted and the local operator performs well whereas it in a situation with less violations the swap may have been rejected. Operators that unallocated an event are very successful with events that cause constraint violations. The learning mechanism is less likely to favour unallocating operators as long as other swaps can be made that reduce the number of constraint violations.

We expect that anticipated results can be found by using different initial solutions (e.g. all events start unallocated) or skewing the learning process or operator selection process in favour of operators that unallocated events. This should reduce both the number of soft constraints violations and achieved frequency.

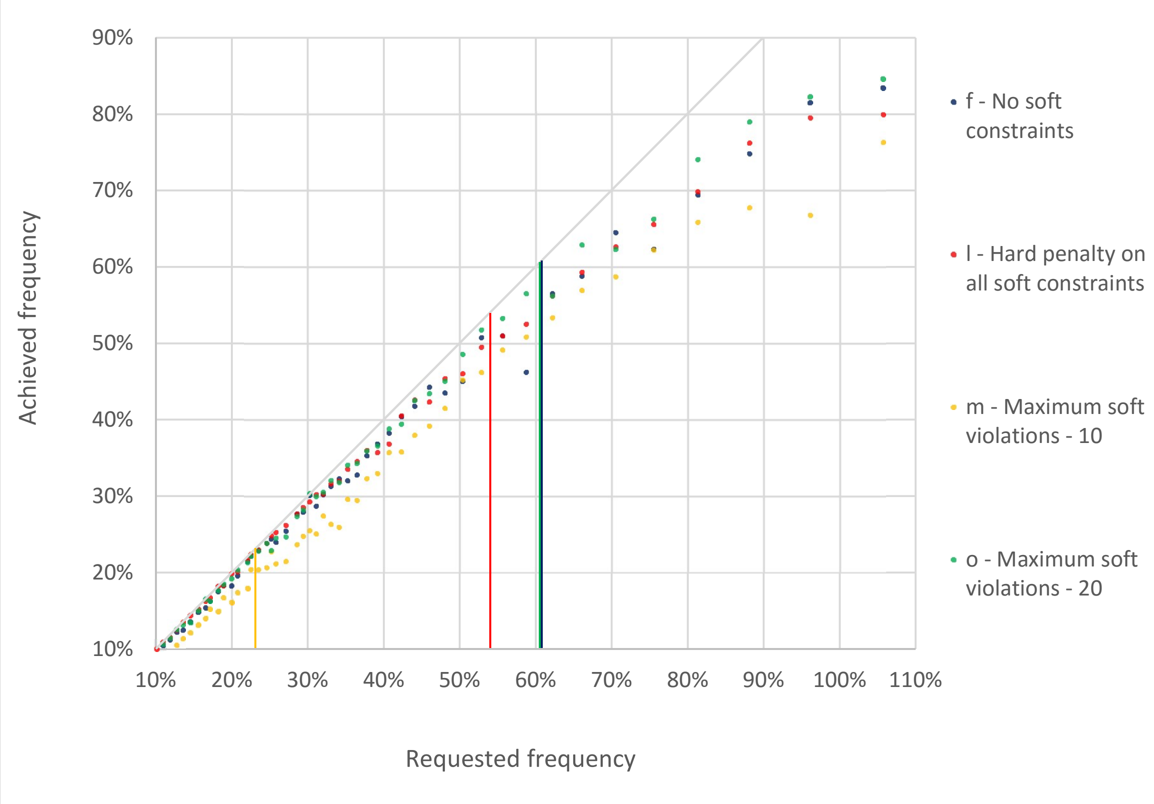


FIGURE 5.4-1 COMPARISON OF CRITICAL UTILISATIONS WITH DIFFERENT METHODS OF AGGERAGATING SOFT PENALTIES

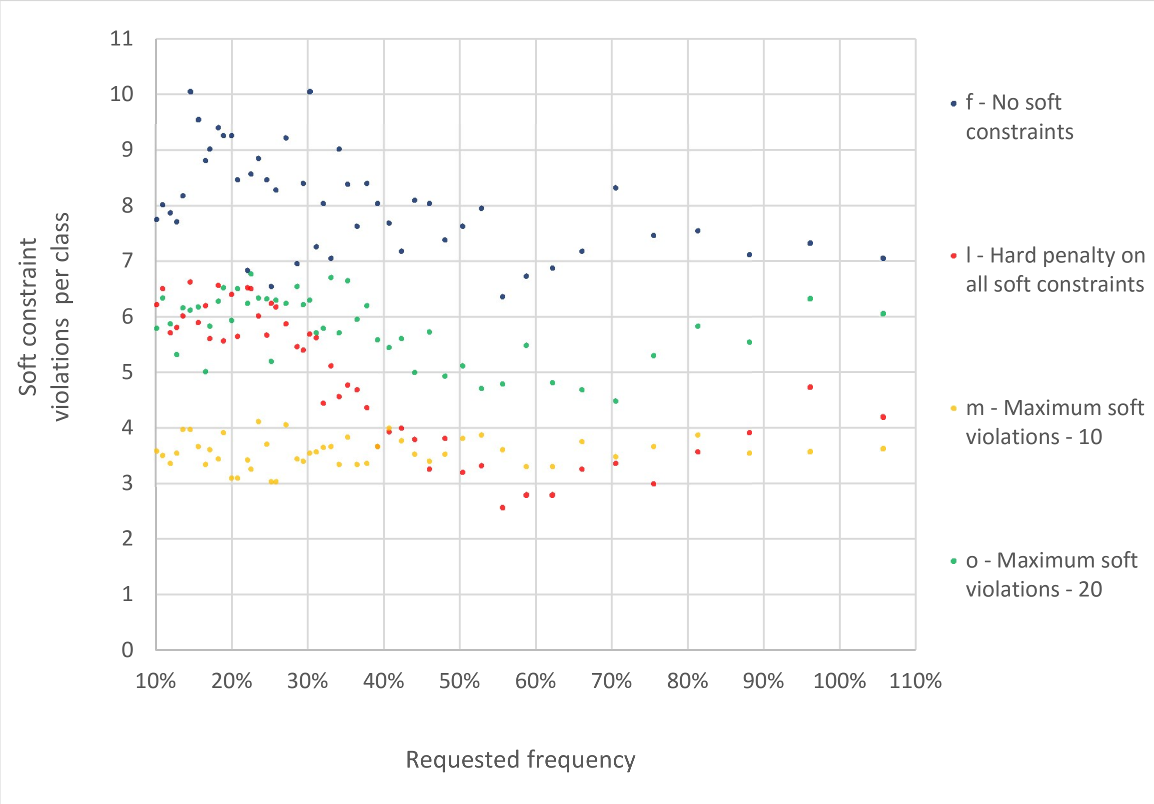


FIGURE 5.4-2 EFFECT OF AGGREGATING SOFT CONSTRAINTS VIOLATIONS ON THE AVERAGE NUMBER OF SOFT CONSTRAINT VIOLATIONS PER CLASS PER WEEK

5.5 SUMMARY

The time during which students can expect events can be reduced drastically. Using just a few timeslots on the middle of the day can only be done as long as there is an overcapacity of space. Splitting classes into a morning and afternoon group or into a Monday-Wednesday-Thursday and Tuesday-Wednesday-Friday-group reduces the time window during which events may be scheduled too, without sacrificing any capacity and without causing scheduling problems due to the extra time constraints.

Improving attendance by clustering events on a few days by using a minimum workload per day has a large negative effect on the critical frequency. The principle was propagated by the students but can cause difficulties with finding a feasible timetable, especially when classes have few events to attend. We will need to schedule by hand or make it as a soft constraint to accommodate this student preference.

When improving timetables by avoiding free periods, spread the workload of a class over a set number of days etc., we can get success without increasing the required space to some extent. However, stacking numerous constraints makes complying to all of them near impossible. It is probably best to start with a high threshold of constraint violations per class (e.g. 20) and lower it each time a timetable is made until no feasible timetable can be produced by the timetablers.

6 SUITABILITY FOR DECISIONMAKING

We have proposed a model that helps to find the demand for educational space given a set of events and a set of scheduling rules. That model has been backed working software for the timetabling part. This chapter describes what can be expected of the model and how to use it.

6.1 SOFTWARE

During the project several pieces of software have been written. These have been written in Visual Basic for Excel so that they can be adjusted and expanded with basic office software. Historic timetables can be exported from Syllabus+ to an Excel file and staff availabilities can be exported from Weper4mens to an Excel file too. Information from these export files can be copied into the timetabling Excel file and used immediately. However, we can also submit our own information (if we want to test several staff/event combinations or try offering courses to multiple classes in the same room etc.).
If the model from this project will be used in the future, it is ready to work with historic data and custom input, written to compare a specific scenario. It can be expanded or refined by anyone with some knowledge of MS Excel.

6.2 HOW-TO-USE

Ideally, we would run the model under a few different constraints and get a plot of requested and achieved utilisations with perfect lines with a sharp bend in it at some point. This ideal is unachievable.
Based on historic timetables and sparring sessions with timetablers, we could determine up to which utilisation or frequency feasible timetables can be made by the scheduling department. With this information one could make the determination of the critical values (below which it is almost always possible to create a feasible timetable) more accurate.
With the results from the model, we can predict whether or not it is worthwhile to try and use certain scheduling rules or if they are unattainable anyway. Furthermore it is worth noting that working with soft and hard constraints shows that scheduling
What the model can do is rank different combinations of policies and scheduling rules on a scale of attainability. The key is to find the point on that scale up to which the organisation can produce feasible timetables that are acceptable to all parties involved.

6.3 IMPROPER USE

This model or the timetabling part of it can be misused. Most parts of the model can be adjusted and can create an unintended bias. A basic knowledge of what is been done during selection processes and determining sequences in the model is essential when making changes.
Another important topic is the penalty structure. It is possible to demand that all events are scheduled in the first four timeslots of a day, and that all classes have a minimal workload of six timeslots per day. These contradicting constraints could lead to a large portion of unallocated events or a lot of violated constraints. Giving the first constraint a low level penalty should lead to a timetable in which most classes have a workload of at least six timeslots per day and most events concentrated in the morning.

Furthermore using an unsuitable initial solution can also create a distorted image. If we know that we want to schedule all events of a class on a single day, we should schedule all events for a class in sequence on a single day. Otherwise we will waste a lot of iterations per run to reduce avoidable penalty occurrence when we could have been searching for the optimal timetable.

6.4 ACCURACY OF THE MODEL

The model gives an indication of what space is needed to enable timetables that comply with a certain set of constraints. However the key aspect, the critical frequency or critical utilisation is determined by eye. This is subject to interpretation, we cannot provide a confidence interval or something similar. It is extremely helpful to match visual cues in the graphs with knowledge of what is achievable from practice.

One of the aspects that is not incorporated in the model is scheduling courses in series over multiple weeks. This is an important aspect in practice, this will limit the timetablers significantly compared to the model that only schedules one week. The model will never grasp the complexity of reality even if it becomes perfectly accurate.

The accuracy of the model can be increased by improving the timetables. There are two main ways to do so: create better initial solutions and increase the number of iterations for each run. The initial solutions used so far took scheduling conflict into account and the room sizes and types too. Creating a constructive heuristic that can cope with the other constraints would improve the starting point of the search. However, this will require extra computation time just like increasing the number of iterations per run. Brute force helps with finding better results but we need huge amounts of computation time to get really small improvements.

6.5 SUMMARY

This chapter revolves around the last research question:

V. *How can the previous solution be made suitable for the support of decision making?*

The model cannot predict exactly how much lecture rooms are needed for the next semester, nor would it be useful to do so. What we can find is what policies have a large negative influence on space usage and if there are other ways to achieve the same goal without increasing the demand for educational space. (E.g. using morning and afternoon classes instead of scheduling all event in timeslot three to ten.)

This project has led to a tool that can help predict the feasibility of introducing new scheduling principles without the need for a team of timetablers to find out by trying. The model is ready to use and prepared for future modification.

7 CONCLUSION

We started this project with the following research question:

How can timetables and space-use be improved by designing a scheduling tool?

The answer we found is: By developing a tool that predicts the feasibility of timetables under given constraints. This enables us to determine if it is worthwhile to try to implement certain scheduling rules or if a reduction of space would lead to serious capacity problems.

I. *Which of the policies in place influence the timetables?*

A lot of binding policies are set by parties outside the cluster Finance & Accounting, e.g. the lecture times, labour regulations, what building and lecture rooms can be used. This leaves little room for timetablers to accommodate additional wishes besides creating feasible timetables with most events between 9:20 and 17:00. However, there are ideas on how to improve the timetables for students and to improve study succes. Some of them seem attainable within the current situation at the Fraijlemaborg.
Timetabling is based on the curricula. Programmes have a lot of freedom in setting the curriculum but scheduling is subordinate to educational considerations. Due to the work involved, curriculum changes are limited to a minimum.
Space and staff capacity is limited by budgets and are hard to change. How staff is deployed follows partly from collective labour agreements and task assignment standards.
However all lot is done based on tacit knowledge. A relatively small team of coordinators works with the programme manager on things like class size, course content and composition and preferred timeslots to teach courses.

II. *What makes a good timetable according to students, teaching staff, scheduling department and school management?*

The preferences of students and staff members with regard to timetabling can be found in section 2.6. Both students and staff indicate that a good timetable stimulates attendance and attentiveness. Therefore course formats on day should vary and events should be scheduled in timeslots favoured by students. This may conflict with the time on task of students, which is important to the staff. Students prefer a compact timetable on a limited number of days; whereas staff prefer students to attend events spread over the week such that they work on their education on all workdays.

III. *What are current bottlenecks in the availability of resources?*

We did not identify a single bottleneck in the eductional logistics. Basically there are enough rooms and available lecturers for the number of events that we try to schedule. Whenever we add a constraint, we make the timetabling process a bit more rigid. Eventually, the whole process grinds to a halt but it is impossible to blame it on one constraint or resource.

Timetablers perceive scheduling difficulties due to low availability of some staff members. Differentiating between necessary and preferred availability may be useful to get some flexibility.

IV. How can user preferences and restricting policies be combined into timetables?

Preferences and proposed policies can be translated into constraints which can be used in a penalty structure which in turn can be used to optimise timetables by using a Simulated Annealing heuristic. We use a multi-tier penalty structure to get the heuristic to schedule as much events as possible without violating set constraints. Finding a way to combine them all or a few of them in one timetable requires some fiddling around with penalty values to prioritise them as it proved to be impossible to comply with all of them at the same time. Applying all constraints at once will result in timetables with only a fraction of the events in them. By using multiple levels of penalties, we can create timetables that comply with certain policies and adhere to preferences as much as possible but not to the point that events remain unallocated.

TABLE 6.5-1 INTERFACE OF THE TOOL WITH MANGERIAL AREAS.

	EDUCATION PLANNING	FACILITIES PLANNING	PERSONNEL PLANNING	FINANCE PLANNING
STRATEGIC	- Setting the curriculum. - Number of lecture days per week. - Class size. - Lecture time per week.	- Real estate. - Number, capacity and type of rooms.	- Workforce size. - Qualifications and skills of the total workforce.	- Central budgets. - Investment plans.
TACTICAL	- Determining course formats.	- Block planning. - Setting business hours. - Equipment in rooms.	- Balancing staff availability in a week - Assigning lectures to courses and classes. - Training staff members.	- Assign general budgets to projects/ departments. - Decision on what is leading performance or budget and to what extent.
OPERATIONAL (OFFLINE)	- Balancing lectures and course formats over the week.	- Assigning lectures to room and timeslot.	- Scheduling classes and other activities per staff member.	- Cost allocation.
OPERATIONAL (ONLINE)	- Adjusting number of lectures to teaching pace.	- Urgent repairs to the rooms and equipment. - Returning unused rooms to the planning pool.	- Replacing absentees.	- Measuring and adjusting expenses.

V. How can the previous solution be made suitable for the support of decision making?

The tool that we have created can help the thought process on how to adress student needs under the given circumstances at the AUAS and can also give an idea of the viability of new timetabling policies. It will hopefully build the confidence within the organisation to try some principles that used to be too far out of the comfort zone.

The managerial areas identified in section 2.1 are shown below in Table 6.5-1. The tool that we developed shows the impact of certain strategic and tactical choices on the processes at an operational level.

8 RECOMMENDATIONS AND FURTHER RESEARCH

During the project we touched on topics that are not included in this thesis, which are shared in the next few paragraphs. We recommend to look into the considerations presented in the first three paragraphs. The last paragraph outlines a research direction that may improve our model and help the timetabling department.

8.1 VARIABLE CLASS SIZE

To get the most out of the available educational space, its occupancy should be maximised by matching class and room size better. However, one of the problems here is that class sizes are not constant. The class sizes, especially in the first year, vary due to students dropping out. Furthermore, not all students in a class will attend every event. Perfect attendance is sometimes optimistically assumed even though it is not realistic.

Teaching certain (parts of) courses to multiple classes at the same moment in the same room reduces the demand for space. If we get a better picture of the number of students that attend events, we can combine events without switching from small-scale education to large-scale education. This large part is important as the maximum number of students at an event is an important part of educational policy.

It may be worthwhile to look more in depth into the decrease of class size per year and programme and regroup the students at set points in time. This can help prevent overcrowded rooms early in the academic year and low occupancy later on. Furthermore keeping track of attendance numbers can help to predict future attendance.

8.2 SPREADING RESOURCE DEMAND

The simplest way to lower required capacity is lowering peak demand. In case of educational space, we have to spread the demand for roomslots over the year.

Per course, the number of lecture hours per week is known and the required type and size of room too. The number of classes that have to take a course is also known. By combining this information for all courses of all programmes, we can balance the total demand for space per week and minimise the required space. It may be useful to let someone explore if curricula can be adjusted such that resource demand is spread evenly over the year without jeopardising precedence relations between and cohesion of courses in a programme.

8.3 INDIVIDUALISATION

We have conducted some interviews and found that staff members link attendance to study success. Students give the timing of courses as the main reason to skip them. The wait between lectures or the travel time is too long or events conflict with activities in their private life.

One solution to increase attendance would be to take a statistically sound survey among students to determine what they want and adjust timetables accordingly. This gives the programme management some control over the spread over the week and the duration of events per day. However, courses can also be offered multiple times per week while students enrol in a course at a certain timeslot. This enables students to attend events when it suits them. Although this might have negative consequences on the time on task of students and the composition of classes.

Letting students pick their courses can give individual students high quality timetables without causing much scheduling problems. Timetablers only have to spread the courses over the week without using numerous scheduling rules.

8.4 CONSTRUCTIVE HEURISTIC

All timetables are made by hand. Even though no formal heuristic is used timetablers seem to follow similar steps: Start with the events with the most constraints attached to them (e.g. a lecturer that only works one day per week) and end with the events with the least limitations.

The courses per lecturer or class, the capacity per room type, event characteristics etc. are known in advance. This information can indicate which events are easy to schedule and which are not and therefore which events should be scheduled first and in which rooms and timeslots preferably. Looking into constructive heuristics can provide timetablers with guidelines or rules of thumb that enables them to work faster and improve their timetables.

9 BIBLIOGRAPHY

Ahmed, L. N., Özcan, E., & Kheiri, A. (2015). Solving high school timetabling problems worldwide using selection hyper-heuristics. *Expert Systems with Applications, 42*(13), 5463-5471. doi:10.1016/j.eswa.2015.02.059

Bai, R., Blazewicz, J., Burke, E. K., Kendall, G., & McCollum, B. (2012). A simulated annealing hyper-heuristic methodology for flexible decision support. *4OR, 10*(1), 43-66. doi:10.1007/s10288-011-0182-8

Bettinelli, A., Cacchiani, V., Roberti, R., & Toth, P. (2015). An overview of curriculum-based course timetabling. *TOP*, 1-37. doi:10.1007/s11750-015-0366-z

Beyrouthy, C., Burke, E. K., Landa-Silva, D., McCollum, B., McMullan, P., & Parkes, A. J. (2008). Threshold effects in the teaching space allocation problem with splitting. *European Journal of Operational Research (EJOR)*.

Beyrouthy, C., Burke, E. K., Landa-Silva, J. D., McCollum, B., McMullan, P., & Parkes, A. J. (2006). Understanding the role of UFOs within space exploitation. *Proceedings of the 6th International Conference on the Practice and Theory of Automated Timetabling, Brno*, 359-362.

Donaldson, T., & Preston, L. E. (1995). The Stakeholder Theory of the Corporation: Concepts, Evidence, and Implications. *The Academy of Management Review, 20*(1), 65-91. doi:10.2307/258887

Executive Board, A. (2011). Begroting 2012. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2012a). Begroting 2013. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2012b). Profiel en prestatieafspraken Hogeschool van Amsterdam. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2013a). Begroting 2014. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2013b). Contacturen in de HvA. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2013c). Programmadocument Slank. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2014a). Begroting 2015. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2014b). Besluiten en implementatiestrategie ontwerpcriteria hoger onderwijs. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2014c). Kwaliteitsstandaarden roostering. Amsterdam: Amsterdam University of Applied Sciences.

Executive Board, A. (2015). Nieuwsgierige Professionals *Instellingsplan 2015-2020*. Amsterdam: Amsterdam University of Applied Sciences.

Facility Services, A. (2013). Huisvestingsplan 2013 - 2023 *Strategisch kader Hogeschool van Amsterdam*. Amsterdam: Amsterdam University of Applied Sciences.

Freeman, R. E. (1984). *Strategic management : a stakeholder approach*.

Hans, E. W., Houdenhoven, M. v., & Hulshof, P. J. H. (2012). A Framework for Healthcare Planning and Control. In R. Hall (Ed.): Springer.

Kalender, M., Kheiri, A., Özcan, E., & Burke, E. (2013). A greedy gradient-simulated annealing selection hyper-heuristic. *Soft Computing, 17*(12), 2279-2292. doi:10.1007/s00500-013-1096-5

Miles, S. (2012). Stakeholder: Essentially Contested or Just Confused? *Journal of Business Ethics, 108*(3), 285-298. doi:10.1007/s10551-011-1090-8

Mitchell, R. K., Agle, B. R., & Wood, D. J. (1997). TOWARD A THEORY OF STAKEHOLDER IDENTIFICATION AND SALIENCE: DEFINING THE PRINCIPLE OF WHO AND WHAT REALLY COUNTS. *Academy of Management Review, 22*(4), 853-886. doi:10.5465/AMR.1997.9711022105

P&O, A. (2013). Leidraad Taaktoewijzing. Amsterdam: Amsterdam University of Applied Sciences.
Pillay, N. (2014). A review of hyper-heuristics for educational timetabling. *Annals of Operations Research*, 1-36. doi:10.1007/s10479-014-1688-1
Van Beek, H. J. M. (2012). Advies bedrijfsvoering *Domein Economie en Bedrijfsvoering*: BMC.

APPENDICES TO:
A CONSTRUCTIVE TOOL TO PREDICT TIMETABLE FEASIBILITY UNDER USER DEFINED CONSTRAINTS.

MASTER THESIS INDUSTRIAL ENGINEERING & MANAGEMENT

12-02-2016

Author:	Stef van den Bedem BSc
	s0192880
Supervisors:	dr. P.C. Schuur
	ir. H. Kroon
	R.W. Puyt MSc MMC
	R.A. Kouthoofd MSc
Programme:	Industrial Engineering & Management
Track:	Production & Logistic Management

CONTENTS

APPENDIX 1 GLOSSARY

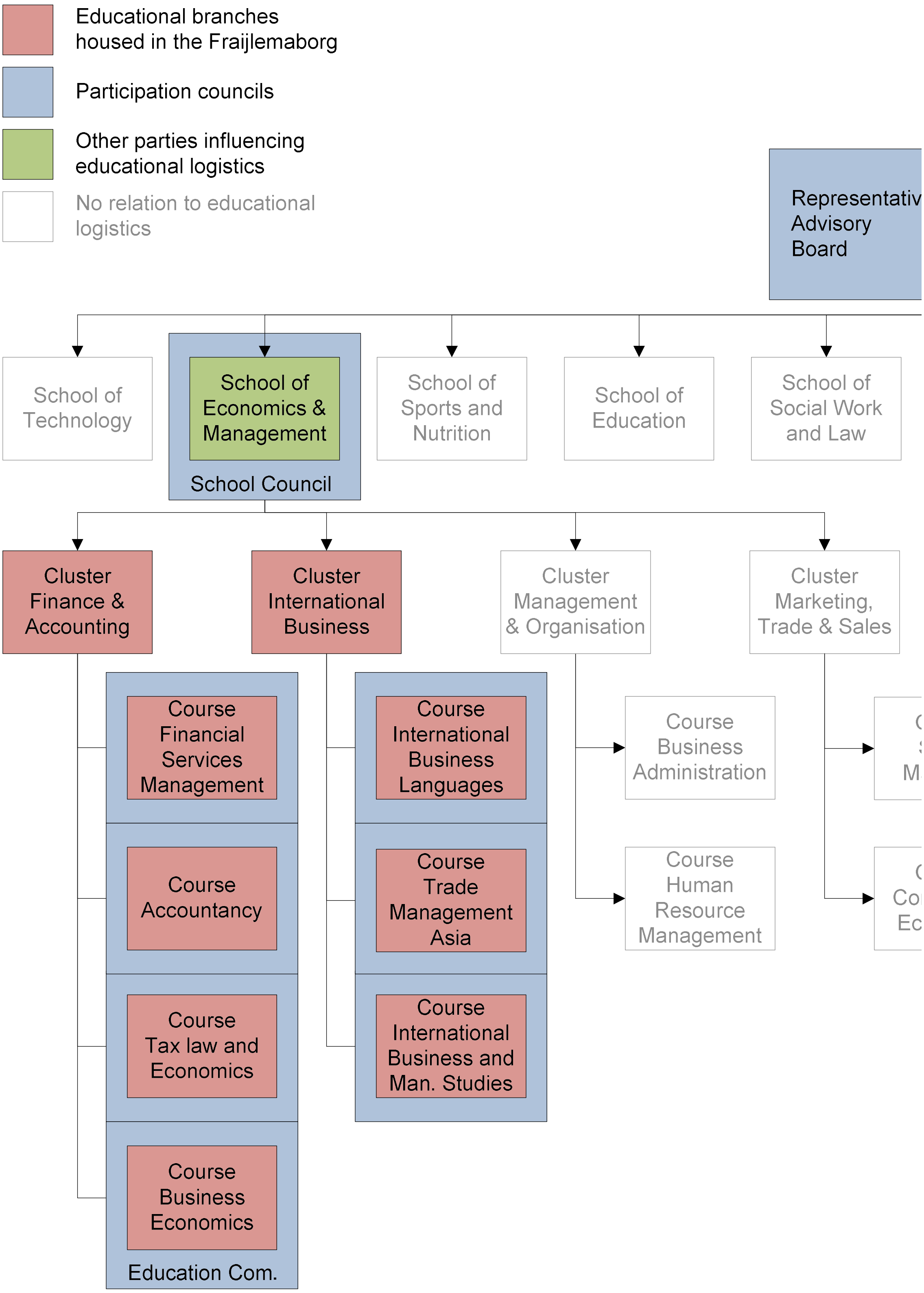
Symbol	Term	Description
a	solution	Solution of a scheduling problem, consisting of an assignment to a time and place for each event
a(e)	assignment	Assignment of event e to a room and timeslot
ã	candidate solution	Solution that may replace the current solution.
a*	optimal solution	The best solution to the scheduling problem
β	cooling factor	Constant that determines how much the temperature is changed each time
c	class	Index used for classes, the set of all possible timeslots is C. c(e) is the sets of classes c that attend event e
c_{acc_all}		Counter for the total number of solutions that is accepted (all operators combined).
c_{acc_o}		Counter for the number of times a solution generated with operator o is accepted.
c_{new_o}		Counter for the number of times operator o generates a new and feasible solution.
c_{tot_o}		Counter for the total number of times operator o is selected.
δ	difference in solution value	Difference between the solutions values of the current solution **a** and candidate solution **ã**
δ(x,y)		Binary variable δ(x,y)=1 if x=y, δ(x,y)=0 if x≠y
e	event	Educational activity, e.g. a lecture or tutorial. The set of all events in the problem is E
	experiment	Evaluation of a scenario. Consists of 40 to 100 runs under the same constraints with a varying requested utilisation or frequency.
F	frequency	Fraction of rooms in a building that is used
F_A	achieved frequency	Fraction of rooms in a building that is used for events that are included in a timetable after solving the scheduling
F_C	critical frequency	Frequency above which it is almost impossible to find a feasible timetable
F_R	requested frequency	Fraction of rooms in a building that is used if all events are included in a timetable
$iteration_c$	current iteration	Indicator of the current iteration of the SA heuristic
$iteration_{max}$	maximum iteration	The number of iterations after which the SA heuristic stops
l	lecturer	Index used for lecturers, the set of all possible timeslots is L. l(e) is the sets of lecturers l that attend event e
lp	learning period	Number of iterations after which the performance of local operators is evaluated and the heuristic learns which operators to use in the coming period.
N(**a**)	neighbourhood	Set of solutions that can be created by applying a local operator to solution **a**.

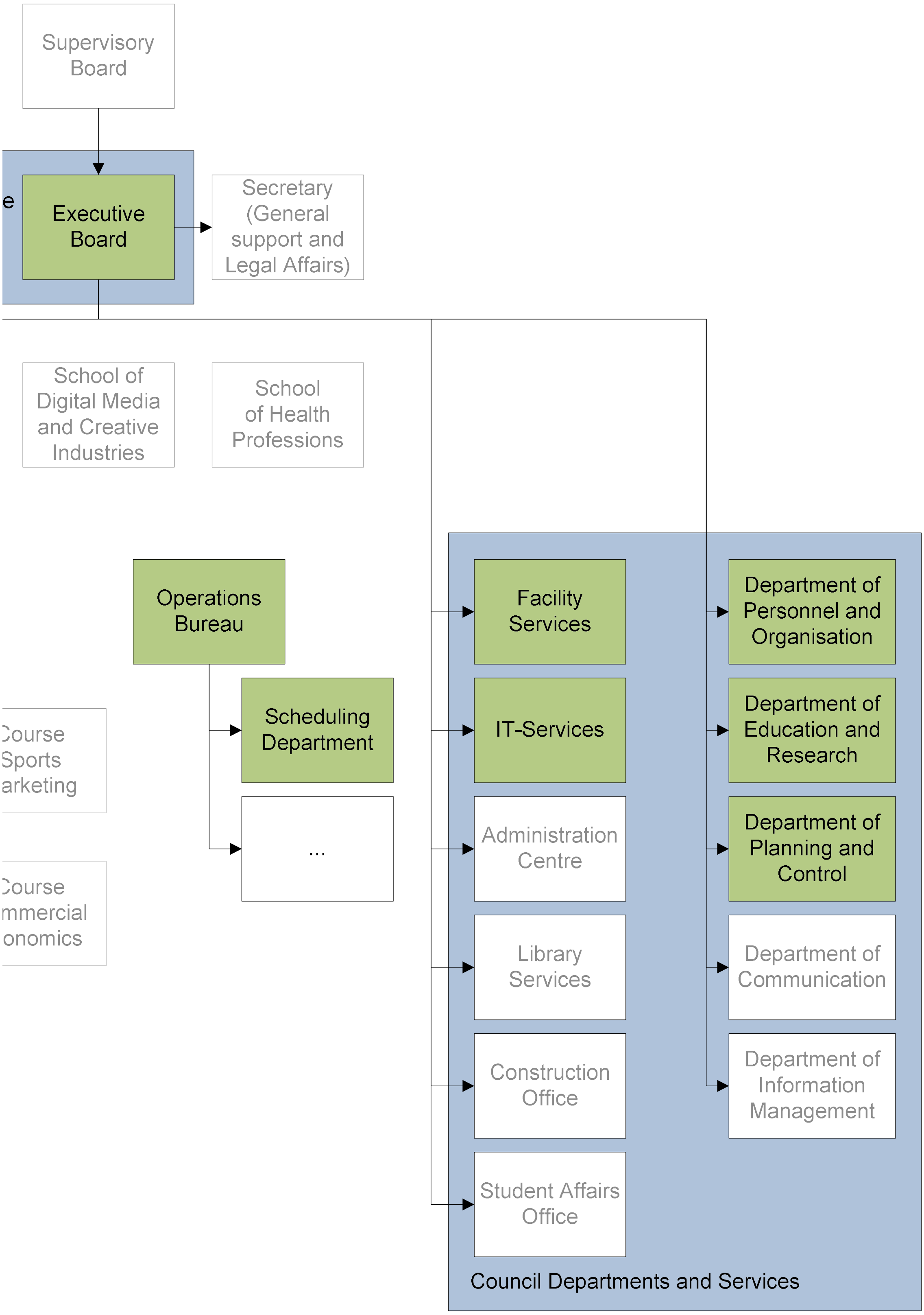
Symbol	Term	Description
n_i	penalty occurence	Constant, the number of times constraint i is violated by a solution.
o	local operator	A low level heuristic that creates candidate solutions by altering the current solution.
O	Occupancy	Fraction of seats-hours in a buidling that is used, limited to roomslots that are used.
P	penalty matrix	Matrix that indicates if a full penalty is awarded for each given room and event type mismatch.
p_i	penalty	Constant, the number of penalty points awarded per violation of constraint i.
pr_o	operator selection probability	Probability that local operator o will be used to create candidate solution **ã**.
π_1 (**a**(e))	room	First variable of the assignment of event e, its location.
π_2 (**a**(e))	timeslot	Second variable of the assignment of event e, its time.
r	room	Index used for rooms, the set of all possible rooms is R.
r_{min}	minimum acceptance ratio	Constant that determines when a reheating period is started.
	roomslot	Measure of capacity of educational space, multiplication of rooms and time.
	run	One instance of solving a scheduling problem by running the SA heuristic.
S	solution space	Set of all possible solutions to a scheduling problem.
	seat-hour	Measure of capacity of educational space, multiplication of seats and time.
t	timeslot	Index used for timeslots, the set of all possible timeslots is T.
$t_{current}$	current temperature	The temperature that is used to determine if a candidate solution is accepted.
t_{end}	end temperature	Constant that will be the final t_current if no reheating phases occur during a run of the heuristic.
t_{impr}	improvement temperature	The last temperature at which a solution was found that was an improvement of the previous solution, the current temperature may revert to this value from time to time.
t_{st}	start temperature	Constant that is used as the initial t_current and as the maximum for t_current while reheating.
U	unavailability matrix	Matrix that indicates if a staff member is available to teach during a timeslot.
U	utilisation	Fraction of seats-hours in a buidling that is used.

Symbol	Term	Description
U_A	achieved utilisation	Fraction of seats-hours in a buidling that is used for events that are included in a timetable after solving the scheduling problem.
U_C	critical utilisation	Utilisation above which it is almost impossible to find a feasible timetable.
U_R	requested utilisation	Fraction of seats-hours in a buidling that is used if all events are included in a timetable.
w_{min}	minimum weight factor	Constant, if the calculated w_o is lower than w_{min}, we use w_{min} instead of w_o.
w_o	weight factor	Weight factor of operator o, used to calculate pr_o.

Acronym	Description
AUAS	Amsterdam University of Applied Sciences
BE	Business Economics, educational programme
CB-CTT	Curriculum based course timetabling
FA	Finance & Accounting, a cluster of programmes within the School Economics & Management
FMB	Fraijlemaborg, the building that houses FA and IBS
HTT	Highschool timetabling
IBS	International Business School, a cluster of programmes within the faculty of Economics & Management
ITC	International Timetabling Competition
SA	Simulated annealing, a type of heursistic
SEM	School of Economics & Management, faculty for research and education in business.

APPENDIX 2 ORGANOGRAM AUAS





Supervisory Board
e
Executive Board
Secretary (General support and Legal Affairs)
School of Digital Media and Creative Industries
School of Health Professions
Operations Bureau
Scheduling Department
...
Course Sports arketing
Course mmercial onomics
Facility Services
IT-Services
Administration Centre
Library Services
Construction Office
Student Affairs Office
Department of Personnel and Organisation
Department of Education and Research
Department of Planning and Control
Department of Communication
Department of Information Management
Council Departments and Services

APPENDIX 3 OVERVIEW OF INTERVIEWS

Nr	Name	Role
1	Robbert Kouthoofd	Lecturer & Education coordinator
2	Jerinca Vreugenhil & Iris Langendijk	Students
3	Henk Tjalsma	Timetabling coordinator
4	Liza Amarchanova & Tonny Cakici	Students
5	Richard Puyt	Lecturer & Education coordinator
6	Demko Bakker	Student
7	Jaap Klouwen	Lecturer
8	Ezrah Bakker	Lecturer & Team coordinator
9	Kees Post	Lecturer
10	Arjen van den Akker	Programme manager Business Economics
11	Pieter Lommerse	Controller cluster Finance & Accounting
12	Teun van Lier	Project worker Educational Logistics (Facility Services)
13	Jeroen Roosen	Staff member Real Estate Management (Facility Services)

APPENDIX 4 OVERVIEW HEURISTIC

This appendix gives an overview of the simulated annealing, the penalty structure and local operators, uninterrupted by descriptions and (except for the local operators) examples. The complete descriptions and example can be found in chapter **Error! Reference source not found.**.

PROBLEM

$E \ = \ \{all\ events\}$
$R \ = \ \{all\ rooms\}$
$T \ = \ \{all\ timeslots\}$
$C \ = \ \{all\ classes\}$
$L \ = \ \{all\ lecturers\}$
$D \ = \ \{all\ days\ of\ the\ week\} = \{1, 2, 3, 4, 5\}$

$d = 1 = \{t_1, t_2 \dots\ t_{12}\}$
$d = 2 = \{t_{13}, t_{14} \dots\ t_{24}\}$
$d = 3 = \{t_{25}, t_{26} \dots\ t_{36}\}$
$d = 4 = \{t_{37}, t_{38} \dots\ t_{48}\}$
$d = 5 = \{t_{49}, t_{50} \dots\ t_{60}\}$

$lunc\ \ (d) = \ \{timeslots\ that\ can\ be\ used\ as\ lunch\ break\}$
$lunc\ \ (1) = \ \{t_6, t_7, t_8\}$
$lunch(2) = \ \{t_{18}, t_{19}, t_{20}\}$
$lunc\ \ \ 3) = \ \{t_{30}, t_{31}, t_{32}\}$
$lunch(4) = \ \{t_{42}, t_{43}, t_{44}\}$
$lunch(5) = \ \{t_{54}, t_{55}, t_{56}\}$

$\boldsymbol{a} = solutio\ \ \ to\ the\ timetabling\ problem$
$\boldsymbol{a}(e) = (r, t) = assignment\ of\ event\ e\ to\ room\ r\ and\ timeslot\ t$
$S \ = \ \{\boldsymbol{a}: \mathbb{F} \rightarrow R\ x\ T\} = solutio\ \ \ space$

$N(\boldsymbol{a}) = neighbour\qquad\quad of\ \boldsymbol{a}$
$N(\boldsymbol{a}) \subseteq S$
$\tilde{\boldsymbol{a}} \in N(\boldsymbol{a})$

$\delta = f(\tilde{\boldsymbol{a}}) - f(\boldsymbol{a})$

c_{tot_o}	*The total number of times operator o is selected.*
c_{new_o}	*The number of times operator o generates a new and feasible solution.*
c_{acc_o}	*The number of times a solution generated with operator o is accepted.*
c_{acc_all}	*The total number of solutions that is accepted (all operators combined).*
w_o	*Weight factor of operator o.*
pr_o	*Probability of selecting operator o.*
$t_{current}$	*The current temperature*
t_{impr}	*The temperature at which the last improving solution was found*
β	*Cooling parameter, determines the speed of cooling and reheating, a constant*
n	*number of iterations per temperature,a constant*

$Accept\ solution\ if$: $\quad \delta < 0$

$or\ if$: $\quad X < e^{-\left(\frac{\delta}{t_{current}}\right)}, \quad X \sim U(0,1)$

$$pr_o = \frac{w_o}{\sum_O w_o}$$

$Select\ operator\ i\ when$: $\quad \sum_{o=1}^{i-1} p_{\ o} < X < \sum_{o=1}^{i} p_{\ i} \qquad X \sim U(0,1)$

$$\beta = \frac{(t_{st} - t_{end}) * n}{t_{st} * t_{end} * iteration_{max}}$$

$After\ n\ iterations$: $\quad t_{current} := \frac{t_{current}}{1 + \beta * t_{current}}$

$When\ reheating\ \ after\ each\ iteration \qquad t_{impr} := \frac{t_{impr}}{1 - \beta * t_{impr}}$

$$t_{current} := t_{impr}$$

while (*iteration*$_c$ < *iteration*$_{max}$)
    iteration$_c$ = *iteration*$_c$ + 1
    pr = 0
    for ∀*o*
        pr := *pr* + *pr*$_o$
        if random(0,1) < *pr* **then**
            generate $\tilde{\boldsymbol{a}}$ with operator *o*
            c_{tot_o} := c_{tot_o} + 1
            exit for-loop
        endif
    next
    if δ <= 0 **and** a new solution was generated **then**
        $\boldsymbol{a}$:= $\tilde{\boldsymbol{a}}$
        t_{impr} := $t_{current}$
        stop reheating

```
                c_new_o   := c_new_o + 1
                c_acc_o   := c_acc_o + 1
                c_acc_all := c_acc_all + 1
        endif
        if δ > 0 and a new solution was generated then
                c_new_o   := c_new_o + 1
                if e^-(δ/t_current) > random(0,1) then
                        a := ã
                        c_acc_o   := c_acc_o + 1
                        c_acc_all := c_acc_all + 1
                endif
        endif
        if reheating then
                t_impr := t_impr / (1 - β * t_impr)
                t_current := t_impr
        endif
        if mod(iteration_c, lp) = 0 then
                if (c_acc_all / lp) < r_min then
                        start reheating
                        t_impr := t_impr / (1 - β * t_impr)
                        t_current := t_impr
                        for ∀o
                                if c_total_o = 0
                                        w_o := w_min
                                else
                                        w_o := max{w_min, c_new_o / c_acc_all}
                                endif
                        next

                else
                        t_current := t_current / (1 + β * t_current)
                        for ∀o
                                if c_total_o = 0
                                        w_o := w_min
                                else
                                        w_o := max{w_min, c_acc_o / c_acc_all}
                                endif
                        next
                endif
                c_tot_o := 0
                c_new_o := 0
                c_acc_o := 0
                c_acc_all := 0
        endif
loop
```

GOAL FUNCTION AND PENALTY STRUCTURE

$$p_i \ = \ score\ per\ penalty\ i$$
$$n_i \ = \ number\ of\ penalties\ i$$
$$f(\boldsymbol{a}(e)) = \sum_{i=1}^{15} p_i * n_i(\boldsymbol{a}(e))$$

$$\delta_{x,y} = 1 \quad if\ x = y$$
$$\quad\ \ = 0 \quad if\ x \neq y$$

$$p_1 * n_1 = p_1 * \sum_R \sum_T \max\left(0\,, \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} \ - 1\right)$$

$$p_2 * n_2 = p_2 * \sum_E \sum_R \sum_T \sum_{i=0}^{group\ size(e)-1} \left(\delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} \ * \delta_{room\ size(r),i}\right)$$

$$p_3 * n_3 = p_3 * \sum_E \sum_R \boldsymbol{P}(type(e), type(r) * \delta_{\pi_1(\boldsymbol{a}(e)),r}$$

$$\boldsymbol{P} = penalty\ of\ assigning\ an\ event\ of\ type\ te\ to\ a\ room\ of\ type\ tr \quad \begin{bmatrix} p_{1,0} & \cdots & p_{te,0} \\ p_{1,1} & \cdots & p_{te,1} \\ \vdots & \ddots & \vdots \\ p_{1,tr} & \cdots & p_{te,tr} \end{bmatrix}$$

$$p_4 * n_4 = p_4 * \sum_R \sum_T \max\left(0\,, 1 - \sum_E \delta_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t}\right)$$

$$p_5 * n_5 = p_5 * \sum_R \sum_T \max\left(0, capacity(r) - \quad {}_{E} \quad {}_{\pi_1(\boldsymbol{a}(e)),r} * \delta_{\pi_2(\boldsymbol{a}(e)),t} * group\ size(e)\right)$$

$$p_6 * n_6 = p_6 * \sum_L \sum_T \max\left(0\,, \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l} \ - 1\right)$$

$$p_7 * n_7 = p_7 * \sum_T \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \boldsymbol{U}(l(e), t)$$
$$\boldsymbol{U}\,(l, t) \ = \ 1 \quad if\ lecturer\ l\ is\ not\ available\ during\ timeslot\ t$$
$$\quad\quad\ \ = \ 0 \quad if\ lecturer\ l\ is\ available\ during\ timeslot\ t$$

$$p_8 * n_8 = p_8 * \sum_L \sum_D \delta_{\sum_{t \in lunch(d)} 1 - \delta_{\left(\sum_E \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}, 3}$$

$$p_9 * n_9 = p_9 * n_9 = \sum_L \sum_D \sum_{i=4+1}^{max\ hrs\ per\ day} \delta_{x,i}$$

$With$:

$$x = \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}\right)\right\} - \max\left\{(first\ timeslot\ on\ d) - 1\ , z * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{z}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{l(e),l}\right)\right),0}\right)\right\}$$

$y = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

$z = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

$$p_{10} * n_{10} = p_{10} * \sum_C \sum_T \max\left(0\ , \sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\ - 1\right)$$

$$p_{11} * n_{11} = p_{11} * \sum_C \max\left(0\ , \sum_{i=12}^{21} \sum_{j=1}^{10} \left(p_i * n_i(c) * \delta_{p_i,j}\right) - 20\right)$$

$$p_{12} * n_{12} = p_{12} * \sum_C \sum_D \delta_{\sum_{t \in lunch(d)} 1 - \delta_{\left(\sum_E \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}, 3}$$

$$p_{13} * n_{13} = p_{13} * \sum_C \sum_D \sum_{i=8+1}^{max\ hrs\ per\ day} \delta_{x,i}$$

$With$:

$$x = \max\left\{ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\} - \max\left\{(first\ timeslot\ on\ d) - 1\ , z * \left(\delta_{\left(\sum_E \sum_{t=fir\quad on\ d}^{z}\left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right)\right\}$$

$y = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

$z = \{first\ timeslot\ on\ day\ d, \dots\ , last\ timeslot\ on\ day\ d\}$

$$p_{14} * n_{14} = p_{14} * \sum_C \sum_D \sum_{i=1}^{min\ hrs\ per\ day - 1} \delta_{(12-x),i}$$

$With$:

$$x = \sum_{t=1}^{12} \delta_{\left(\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right),0}$$

$$p_{15} * n_{15} = p_{15} * \sum_C \sum_D \Bigg(\max\Big\{ 0, \text{intended } start\ time\ (d) \\ - \max\Big\{ first\ timeslot\ on\ d, 1 + \ x * \delta_{\left(\sum_E \sum_{t=first\ on\ d}^{x} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0} \Big\}\Big\} \\ + \ \max\Big\{ 0, \max\Big\{ \ y * \left(1 - \delta_{\left(\sum_E \sum_{t=y}^{last\ on\ d} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right) \Big\} \\ - \ intended\ end\ time(d) \Big\}\Bigg)$$

$With$:
$x = \{first\ timeslot\ on\ day\ d, \ldots\ , last\ timeslot\ on\ day\ d\}$
$y = \{first\ timeslot\ on\ day\ d, \ldots\ , last\ timeslot\ on\ day\ d\}$

$$p_{16} * n_{16} = p_{16} * \sum_{C \in morning\ group} \sum_D \max\Big\{ 0, \max\Big\{ \ x * \left(1 - \delta_{\left(\sum_E \sum_{t=x}^{last\ on\ d} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right) \Big\} \\ - seventh\ timeslot(d \ \Big\}$$

$With$:
$x = a\ timeslot\ between\ the\ seventh\ and\ the\ last\ timeslot\ on\ day\ d.$

$$p_{17} * n_{17} = p_{17} * \sum_{C \in morning\ group} \sum_D \Bigg(\max\Big\{ 0, seventh\ timeslot\ on\ d \\ - \ \max\Big\{ first\ timeslot\ on\ d, 1 + \ x * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{x} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0} \right) \Big\}\Big\}\Bigg)$$

$With$:
$x = a\ timeslot\ between\ the\ first\ and\ the\ seventh\ timeslot\ on\ day\ d.$

$$p_{18} * n_{18} = p_{18} * \sum_C 1 - \left(1 - \delta_{\left(\sum_E \sum_{t \in d = monday} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right)\right),0}\right) \\ * \left(1 - \delta_{\left(\sum_E \sum_{u \in d = friday} \left(\delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c}\right)\right),0}\right)$$

$$p_{19} * n_{19} = p_{19} * \sum_C \left(1 - \delta_{\left(\sum_D \left(1 - \delta_{\left(\sum_E \sum_{t \in d} \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}\right),0}\right)\right), predefined\ number\ of\ days} \right)$$

$$p_{20} * n_{20} = p_{20} * \sum_{C} \sum_{D} \left(\sum_{t=x}^{\min(lunch(d))-1} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}, 0} + \sum_{u=1+\mathrm{m}\quad(lunch(d))}^{y} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c}, 0} \right)$$

$$x = \max\left\{ (first\ timeslot\ on\ d), 1 + z * \left(\delta_{\left(\sum_E \sum_{t=first\ on\ d}^{z} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c} \right)\right), 0} \right) \right\}$$

$$y = \max\left\{ z * \left(1 - \delta_{\left(\sum_E \sum_{t=z}^{last\ on\ d} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c} \right)\right), 0} \right) \right\}$$

$$z = \{first\ timeslot\ on\ day\ d, \dots, last\ timeslot\ on\ day\ d\}$$

$$p_{21} * n_{21} = p_{21} * \sum_{C} \sum_{D} \left(\sum_{t=x}^{\min(lunc\quad(d))-1} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c}, 0} * \left(1 - \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),t-1} * \delta_{c(e),c}, 0} \right) + \sum_{u=1+\max(lun\quad(d))}^{y} \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u} * \delta_{c(e),c}, 0} * \left(1 - \delta_{\sum_E \delta_{\pi_2(\boldsymbol{a}(e)),u-1} * \delta_{c(e),c}, 0} \right) \right)$$

$With$:

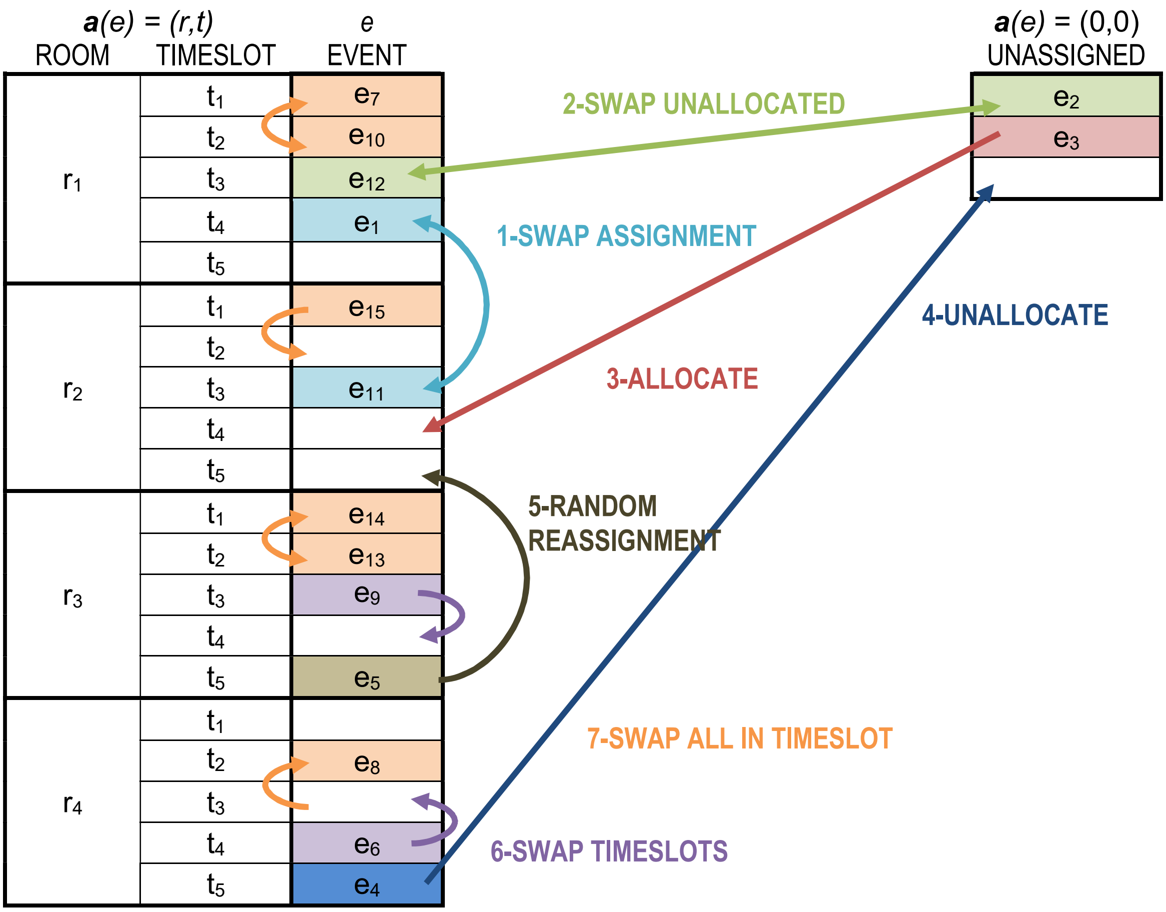
$$x = \max\left\{ (first\ timeslot\ on\ d), 1 + z * \left(\delta_{\left(\sum_E \sum_{t=fir\quad on\ d}^{z} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c} \right)\right), 0} \right) \right\}$$

$$y = \max\left\{ z * \left(1 - \delta_{\left(\sum_E \sum_{t=z}^{last\ on\ d} \left(\delta_{\pi_2(\boldsymbol{a}(e)),t} * \delta_{c(e),c} \right)\right), 0} \right) \right\}$$

$$z = \{first\ timeslot\ on\ day\ d, \dots, last\ timeslot\ on\ day\ d\}$$

LOCAL OPERATORS

$$E = \{all\ events\} = \{e_1, e_2, e_3 \dots e_{15}\}$$
$$R = \{all\ rooms\} = \{r_1, r_2, r_3, r_4\}$$
$$T = \{all\ timeslots\} = \{t_1, t_2, t_3, t_4, t_5\}$$



Operator 1 – Swap two

Two rooms are randomly selected and in each room an event is randomly chosen. The rooms and timeslots of these events are swapped. In case we select room r_1 and r_2 and then select two events, e_1 and e_{11}, at timeslots t_4 and t_3 respectively, $\tilde{\boldsymbol{a}}$ is the same as $\boldsymbol{a}$ except for $\boldsymbol{a}(e_1)$ and $\boldsymbol{a}(e_{11})$.

$$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) = (r, t | r = r_1) \;\; \rightarrow \;\; \boldsymbol{a}(e_a) = \boldsymbol{a}(e_1) = (r_1, t_4)$$
$$\boldsymbol{a}(e_b) \in \boldsymbol{a}(e) = (r, t | r = r_2) \;\; \rightarrow \;\; \boldsymbol{a}(e_b) = \boldsymbol{a}(e_{11}) = (r_2, t_3)$$

$$\tilde{\boldsymbol{a}}(e_5) = (r_2, t_3)$$
$$\tilde{\boldsymbol{a}}(e_7) = (r_1, t_4)$$

Operator 2 – Swap unallocated

One of the unallocated and one allocated events are randomly selected and swapped. The room and timeslot remain unaltered.

$$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) \neq (0,0) \;\; \rightarrow \;\; \boldsymbol{a}(e_a) = \boldsymbol{a}(e_{12}) = (r_1, t_3)$$
$$\boldsymbol{a}(e_b) \in \boldsymbol{a}(e) = (0,0) \;\; \rightarrow \;\; \boldsymbol{a}(e_a) = \boldsymbol{a}(e_2)$$

$\tilde{\boldsymbol{a}}(e_5) = (0,0)$
$\tilde{\boldsymbol{a}}(e_6) = (r_1, t_3)$

Operator 3 - Allocate
An unallocated event, a room and timeslot are randomly selected. The event is added to the timetable on that time and place.
$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) = (0,0) \;\; \rightarrow \;\; \boldsymbol{a}(e_a) = \boldsymbol{a}(e_3)$
$(r|\, r \neq 0) \;\; \rightarrow \;\; r_2$
$(t|\, t \neq 0) \;\; \rightarrow \;\; t_5$

$\tilde{\boldsymbol{a}}(e_a) = (r_2, t_5)$

Operator 4 - Unallocate
Un-allocate a randomly selected allocated event.
$\boldsymbol{a}(e_a) \in \boldsymbol{a}(e) \neq (0,0) \;\; \rightarrow \;\; \boldsymbol{a}(e_a) = \boldsymbol{a}(e_4) = (r_4, t_5)$

$\tilde{\boldsymbol{a}}(e_7) = (0,0)$

Operator 5 – Random reassignment
Select a random event and schedule it on a randomly selected place and time.
$e \in E \;\; \rightarrow \;\; e = 5$
$(r|\, r \neq 0) \;\; \rightarrow \;\; r_2$
$(t|\, t \neq 0) \;\; \rightarrow \;\; t_4$

$\tilde{\boldsymbol{a}}(e_5) = (r_2, t_4)$

Operator 6 – Swap timeslot of two
Select randomly select two events and swap their timeslots (and room if one of the events is not assigned to a timeslot).

$e_a \in E \;\; \rightarrow \;\; e = 6$
$e_b \in E \;\; \rightarrow \;\; e = 9$
$\boldsymbol{a}(e_6) = (r_4, t_4)$
$\boldsymbol{a}(e_9) = (r_3, t_3)$

$\tilde{\boldsymbol{a}}(e_6) = (r_4, t_3)$
$\tilde{\boldsymbol{a}}(e_9) = (r_3, t_4)$
$e_a \in E \;\; \rightarrow \;\; e = 6$
$e_b \in E \;\; \rightarrow \;\; e = 3$
$\boldsymbol{a}(e_6) = (r_4, t_4)$
$\boldsymbol{a}(e_3) = (0,0)$

$\tilde{\boldsymbol{a}}(e_6) = (0,0)$
$\tilde{\boldsymbol{a}}(e_3) = (r_4, t_4)$

Operator 7 – Swap all in timeslots

Select two timeslots (t_1 and t_2) and swap the timeslots of all events assigned to either timeslot.

$\boldsymbol{a}(e_a)\forall\boldsymbol{a}(e) = (r, t|t = t_1)$
$\boldsymbol{a}(e_b)\forall\boldsymbol{a}(e) = (r, t|t = t_2)$

$\tilde{\boldsymbol{a}}(e_a) = (r, t_2)$
$\tilde{\boldsymbol{a}}(e_b) = (r, t_1)$

In case an assignment is impossible (the kind of events needed for a swap do not exist, or an event that takes multiple timeslots is split over multiple days), the reassignment is cancelled. Otherwise it is marked as a new candidate solution and it is either accepted or rejected. The current solution is not changed when a candidate solution $\tilde{\boldsymbol{a}}$ is not accepted.

APPENDIX 5 TIMETABLES FOR EXAMPLES PENALTY CALCULATION

e	Type(*e*)
Accountancy 1	1
French 1	1
French 1 Tutorial	3
German 1	1
German 1 Tutorial	3
Physical Education 1	5
Spanish 1	1
Spanish 1 Tutorial	3
Statistics 1	1
Statistics 1 Tutorial	3

r	Type(*r*)	Size(*r*)
FMB 001	1	32
FMB 002	1	28
FMB 003	3	32
FMB Extern	5	50

c	Size(*c*)
1Ca	31
1Cb	31
1Cc	31
Multiple classes	Max{31,31,31}

$T=\{t_1, t_2 \dots t_{30}\}$
D={Monday, Tuesday, Wednesday, Thursday, Friday}

Monday = $\{t_1, t_2 \dots t_6\}$
Tuesday = $\{t_7, t_8 \dots t_{12}\}$
…

lunch(Monday) = t_3
lunch(Tuesday) = t_9
…

t_{start}(Monday) = t_2, t_{end}(Monday) = t_5
t_{start}(Monday) = t_8, t_{end}(Monday) = t_{11}
…

$$\boldsymbol{P} = penalty\ of\ assigning\ an\ event\ of\ type\ te\ to\ a\ room\ of\ type\ tr = \begin{bmatrix} p_{1,0} & \cdots & p_{te,0} \\ p_{1,1} & \cdots & p_{te,1} \\ \vdots & \ddots & \vdots \\ p_{1,tr} & \cdots & p_{te,tr} \end{bmatrix}$$

$$\boldsymbol{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0.2 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

TIMETABLE OF CLASS 1C

<table>
<tr><th></th><th colspan="2">MONDAY</th><th>TUESDAY</th></tr>
<tr><td>9:00-
10:30</td><td></td><td rowspan="3">Accountancy 1
FMB002
1Ca 1Cb 1Cc
Lawson</td><td rowspan="2">Statistics 1
FMB 001
1Ca 1Cb 1Cc
Mathews</td></tr>
<tr><td>10:30-
12:00</td><td></td></tr>
<tr><td>12:00-
13:30</td><td rowspan="3">Physical
Education 1
FMB Extern
1Ca 1Cv 1Cc
Pearce</td><td>t_9</td></tr>
<tr><td>13:30-
15:00</td><td></td><td>t_{10}</td></tr>
<tr><td>15:00-
16:30</td><td></td><td rowspan="2">Spanish 1
FMB 001
1Cc
Panadero</td></tr>
<tr><td>16:30-
18:00</td><td colspan="2">t6</td></tr>
</table>

<table>
<tr><th colspan="2">WEDNESDAY</th><th>THURSDAY</th><th colspan="3">FRIDAY</th></tr>
<tr><td colspan="2">t_{13}</td><td>t_{19}</td><td colspan="3">t_{25}</td></tr>
<tr><td colspan="2" rowspan="2">Statistics 1 Tutorial
FMB 001
1Ca 1Cb 1Cc
Mathews</td><td>t_{20}</td><td rowspan="4">Spanish 1 Tutorial
FMB 001
1Cc
Panadero</td><td rowspan="4">German 1 Tutorial
FMB 001
1Ca
Bäcker</td><td rowspan="4">French 1 Tutorial
FMB 001
1Cb
Boulangier</td></tr>
<tr><td>t_{21}</td></tr>
<tr><td colspan="2">t_{16}</td><td>t_{22}</td></tr>
<tr><td rowspan="2">German 1
FMB 001
1Ca
Bäcker</td><td rowspan="2">French 1
FMB 001
1Cb
Boulangier</td><td>t_{23}</td></tr>
<tr><td>t_{24}</td><td colspan="3">t_{30}</td></tr>
</table>

TIMETABLE OF LECTURER

	MONDAY		TUESDAY	
9:00-10:30	Preperation Internship FMB 001 Bäcker 4A		Professional Skills FMB 001 Bäcker 2B	
10:30-12:00				
12:00-13:30				
13:30-15:00		Decision Making FMB 001 Bäcker 3A		Decision Making FMB 003 Bäcker 3A
15:00-16:30				
16:30-18:00	t6		t_{12}	

WEDNESDAY	THURSDAY	FRIDAY	
t_{13}	~~t_{19}~~		Professional Skills FMB 001 Bäcker 2A
t_{14}	~~t_{20}~~	German 1 Tutorial FMB 003 Bäcker 1Bb, 1Ca	
t_{15}	~~t_{21}~~		
t_{16}	~~t_{22}~~		
German 1 FMB 001 Bäcker 1Bb, 1Ca	~~t_{23}~~		
	t_{24}	t_{30}	

APPENDIX 6 EXPERIMENTS

This appendix gives an overview of the penalty structures used in the experiments. The table on the next page shows the penalties per constraint for each experiment. The penalty for constraint 3 is always 1000 but the penalty matrix for event and room types is different, indicated with either 1000[1] or 1000[2]. In the first case, we only penalise internal events at external locations and external events in the Fraijlemaborg. In the second case, only match events in a room of a matching type or unallocated events do not get a full penalty.

$$\boldsymbol{P} = penalty\ of\ assigning\ an\ event\ of\ type\ te\ to\ a\ room\ of\ type\ tr = \begin{bmatrix} p_{1,0} & \cdots & p_{te,0} \\ p_{1,1} & \cdots & p_{te,1} \\ \vdots & \ddots & \vdots \\ p_{1,tr} & \cdots & p_{te,tr} \end{bmatrix}$$

$$\boldsymbol{P_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$\boldsymbol{P_2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Two constraints have an input value that is not the same for all experiments: constraints 11 and 15.

Constraint 11	X	Y	
m	10		The total penalty score for violations of soft constraints at most X for each class. (Above this threshold a penalty of 1000 is added for each penalty point that exceeds it.)
n	10		
o	20		
p	20		
Constraint 15	X	Y	
f	1	12	All events must be scheduled during timeslot X to Y.
g	2	11	
h	3	10	

	Constraint	Experim. X	Experim. Y	a All pr.	b All pr.	c All pr.	d All pr.
1	Rooms are used for at most one event per timeslot			1000	1000	1000	1000
2	Rooms are large enough to seat all attending students.			0	1000	1000	0
3	Events are scheduled in rooms of a suitable type.			1000[1]	1000[2]	1000[2]	1000[1]
4	Each room is used during each timeslot.			250	0	0	250
5	All seats in a room are used each timeslot			0	10	10	0
6	Lecturers teach at most one event per timeslot			1000	1000	0	1000
7	Lecturers do not teach during timeslots during which they are not available.			0	0	0	1000
8	Lecturers have at least one free timeslot among timeslots X through Y each day.	6	8	0	0	0	1000
9	Events of a lecturer do not span more than X timeslots each day	10		0	0	0	1000
10	Classes attend at most one event per timeslot			1000	1000	0	1000
11	The number of violations of the constraints below is at most X for each class.			0	0	0	0
12	Classes have at least one free timeslot among timeslots X through Y each day.	6	8	0	0	0	0
13	Events of a class do not span more than X timeslots each day	8		0	0	0	0
14	A class has either no or X or more timeslots with events on a day	4		0	0	0	0
15	Events are scheduled in timeslot X through Y each day	3	10	0	0	0	0
16	Events of morning classes are scheduled in timeslot X through Y each day	1	7	0	0	0	0
17	Events of afternoon classes are scheduled in timeslot X through Y each day	7	12	0	0	0	0
18	A class has events on Monday and Friday			0	0	0	0
19	A class' courses are spread over X days	4		0	0	0	0
20	A class has no free timeslots between classes on the same day			0	0	0	0
21	A class has a minimum number of alternations between events and free timeslots. (XXOOOXX is better than XOXOXOX)			0	0	0	0

	Constraint	Experim. X	Y	*e* BE	*f* BE	*g* BE	*h* BE
1	Rooms are used for at most one event per timeslot			1000	1000	1000	1000
2	Rooms are large enough to seat all attending students.			0	0	0	0
3	Events are scheduled in rooms of a suitable type.			1000[1]	1000[2]	1000[2]	1000[2]
4	Each room is used during each timeslot.			250	250	250	250
5	All seats in a room are used each timeslot			0	0	0	0
6	Lecturers teach at most one event per timeslot			1000	1000	1000	1000
7	Lecturers do not teach during timeslots during which they are not available.			0	0	0	0
8	Lecturers have at least one free timeslot among timeslots X through Y each day.	6	8	0	1000	1000	1000
9	Events of a lecturer do not span more than X timeslots each day	10		0	1000	1000	1000
10	Classes attend at most one event per timeslot			1000	1000	1000	1000
11	The number of violations of the constraints below is at most X for each class.			0	0	0	0
12	Classes have at least one free timeslot among timeslots X through Y each day.	6	8	0	0	0	0
13	Events of a class do not span more than X timeslots each day	8		0	0	0	0
14	A class has either no or X or more timeslots with events on a day	4		0	0	0	0
15	Events are scheduled in timeslot X through Y each day	3	10	0	0	1000	1000
16	Events of morning classes are scheduled in timeslot X through Y each day	1	7	0	0	0	0
17	Events of afternoon classes are scheduled in timeslot X through Y each day	7	12	0	0	0	0
18	A class has events on Monday and Friday			0	0	0	0
19	A class' courses are spread over X days	4		0	0	0	0
20	A class has no free timeslots between classes on the same day			0	0	0	0
21	A class has a minimum number of alternations between events and free timeslots. (XXOOOXX is better than XOXOXOX)			0	0	0	0

	Experim.		*i*	*j*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	*s*
	X	Y	BE	BE	BE	BE	BE	BE	BE	BE	All pr.	BE	All pr.
1			1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
2			0	0	0	0	0	0	0	0	0	0	0
3			1000^2	1000^2	1000^2	1000^2	1000^2	1000^2	1000^2	1000^2	1000^2	1000^1	1000^1
4			250	250	250	250	250	250	250	250	250	250	250
5			0	0	0	0	0	0	0	0	0	0	0
6			1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
7			0	0	0	0	0	0	0	0	0	0	0
8	6	8	1000	1000	1000	1000	1000	1000	1000	1000	1000	0	0
9	10		1000	1000	1000	1000	1000	1000	1000	1000	1000	0	0
10			1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
11			0	0	0	0	1000	1000	1000	1000	0	0	0
12	6	8	0	0	0	1000	2	2	2	2	0	0	0
13	8		0	0	0	1000	4	4	4	4	0	0	0
14	4		0	1000	1000	1000	4	4	4	4	1000	0	0
15	3	10	0	0	0	0	0	0	0	0	0	0	0
16	1	7	1000	1000	0	0	0	1000	0	1000	0	0	0
17	7	12	1000	1000	0	0	0	1000	0	1000	0	0	0
18			0	0	0	1000	2	2	2	2	0	0	0
19	4		0	0	0	1000	2	2	2	2	0	0	0
20			0	0	0	1000	1	1	1	1	0	0	0
21			0	0	0	1000	1	1	1	1	0	0	0