Maastricht University

Department of Data Science and Knowledge Engineering

Master's Program in Artificial Intelligence
and Data Science for Decision Making

Master Research Project - Project Plan

Group 18

March 20, 2020

Timetable Scheduling for the Department of Data Science and Knowledge Engineering

Authors:
Yu Fee Chan
Paul Disbechl
Daniel Kaestner
Camilla Lummerzheim
Le Duc Huy Ngo
Guillermo Quintana Pelayo

Supervisor:
Matúš Mihalák

Coordinator:
Kurt Driessens

Contents

1 Context and Motivation

Creating university timetables can be regarded as a specific type of a scheduling problem [9] where jobs are assigned to resources at particular times. In timetable scheduling, a given set of lectures is assigned to rooms and time slots. The aim is to develop a timetable, which defines the time and place of a lecture, that fulfills given constraints. Primarily, these constraints are formulated as a set of hard constraints, such as the availability of lecturers and the required number of lectures per period which have to be satisfied to make the timetable feasible. This also includes not having overlaps, as for example a single lecturer or a room can not be assigned to two courses at the same time. Apart from that, a number of soft constraints or "features" are taken into account as well. Unlike the aforementioned hard constraints, soft constraints determine the quality of a feasible timetable and can be violated. Soft constraints originate mainly from the preferences of students and lecturers, who would for example not want an empty 2-hour block between lectures.

Although "first attempts at computer-aided timetabling were made in the early 60's when universities obtained computers" [1], and it appears to be a trivial optimization problem, the topic is still heavily researched. This is evidenced by regular competitions like the International Timetabling Competition (ITC) [5] and conferences like the International Series of Conferences on the Practice and Theory of Automated Timetabling (PATAT) [8], which was held roughly biannually from 1995 until 2020.

Today, setting up timetables at universities still requires a lot of time and is an intricate task for the person in charge. What is it that makes this problem so complex to solve? One important factor here is the large scale of time slots, courses, lecturers and resources but also the number of constraints. Finding and formulating these constraints is not a trivial task. This is especially true regarding soft constraints, everyone involved could probably name a long list of preferences which contradict those of others. While some people prefer to have all of their courses concentrated on a selected number of days, others would rather have all their courses spread out as much as possible. Similarly, some people may only want to have courses in the early morning, while others only want to have their courses scheduled in the late afternoon. These individual preferences make evaluating a timetable even more difficult, and may lead to a set of schedules which all achieve the same score for a given evaluation metric.

The main difficulty of our proposed research project is that the concrete constraints may vary a lot depending on the structure of the institution, all of which make it nearly impossible to develop a universal timetabling scheduling tool. The Department of Data Science and Knowledge Engineering (*DKE*) is a prime example where a number of students are of the opinion that the schedules could be improved. Currently, these schedules are created by hand. Potentially, the findings of this research could be used to create an automated timetabling tool, which can assist the scheduler in constructing timetables that satisfy the needs of the students at DKE. Additionally, it would be useful to see if and by how much the schedules can be improved using the proposed methods, compared to the schedules generated by the open-source timetabling tool UniTime [10], which has won several ITC competitions [6].

Several specific advantages arise from limiting the scope to timetabling at DKE. The first benefit of this limitation is that we are in direct contact with the Student Affairs Office of the department, where schedules are currently built manually. The second advantage is the small size of the department and the relatively large number of compulsory courses. Especially in the Bachelor program, with the exception of a few number of electives, every student has to take the same courses. Therefore, it is crucial but also easier to avoid an overlap between lectures as there are only three courses per teaching period. As a third advantage, we consider that all of us are studying at DKE. Hence, we know the current schedules and can compare the results of our algorithms with the prior schedules and our own experience. Additionally, as our group is fairly diverse, most of us have different preferences regarding the "optimal" schedule which may lead to a variety of schedules.

This project plan is structured as follows: First, the main objectives and research questions of the project are presented in Section 2. Section 3 then provides an overview about the current state of the art. In Section 4 we describe our approach to the problem and how we plan to design several algorithms to create schedules for DKE. The deliverables are outlined in Section 5, along with our prospective plans regarding time management in Section 6. Section 7 concludes by covering potential risks and contingency plans.

2 Objectives and Research Questions

This project aims to research and compare the performance of three algorithms in creating timetables at DKE. These algorithms are a greedy approach, an Integer Linear Programming (*ILP*) approach and a metaheuristic, such as a genetic algorithm. An overview of the latter two algorithms can be found in Section 3. If new interesting algorithms are found during the implementation of these three, we will also consider their inclusion to the framework. With this goal in mind, we aim to answer the following research questions:

- How can the current schedules at DKE be improved in terms of student satisfaction and construction time?
- How can one choose which soft constraints are more important than others at DKE?
- Which features of a schedule are important to students and how should the constraints be weighted?
- What is a good evaluation metric for comparing timetables, and how should the cost functions be defined?
- How do the different algorithms compare in terms of their runtime and regarding the quality of the schedules they generate?
- How do the results compare to those generated for DKE by a more general open-source timetabling tool like UniTime?

3 State of the Art and Previous Research

Generally, the problem of university timetabling can be seen as finding a way to assign events (lectures, meetings, etc.), to a set of time slots of the resources (lecture rooms). As mentioned in Section 1, this task is subject to a set of hard and soft constraints. Cooper and Kingston have shown that the timetabling problem belongs to a family of NP-complete problems [2]: Taking different well-known NP-complete problems, they use polynomial bounded transformations to reduce them to timetabling problem variants.

Moreover, depending on the imposed constraints by the university and other constraints, the set of feasible timetables can become very small or even empty. For this reason, the algorithms have to allow for some slack which eventually allows constraints to be violated.

Over the years, various automated timetabling approaches have been proposed. Because of the nature of this problem, most of these approaches consider the use of approximation algorithms, which try to get as close as possible to the optimal solution using a small runtime. Specifically, there has been an increase in the use of metaheuristics, which are essentially algorithmic frameworks that can be adapted to be suitable for different optimization problems. Next to these, in more exact approaches, the problem is formulated as an ILP, which is then solved using solvers such as Gurobi or CPLEX.

In a survey performed by Lewis [7], the author concentrates on the five main paradigms of metaheuristics: Ant Colony Optimization, Evolutionary Algorithms, Local Search, Tabu Search, and Simulated Annealing. Specifically, they focus on the different ways in which these algorithms can be used to distinguish and handle certain hard and soft constraints. The algorithms are separated into three categories: one-stage optimization, two-stage optimization, and algorithms that allow for constraint relaxation. It is shown that certain approaches have a better performance depending on the user requirements, and that there is no universally superior algorithm. For this reason, we aim to find approaches that suit the scheduling of DKE courses the best.

Regarding ILP, Daskalaki et al. present a $0-1$ ILP formulation in a case study for the timetabling problem, as seen in many universities [4]. They show that solvable and flexible models can automatize the timetabling process. In another paper, Daskalaki et al. present a two-stage relaxation approach [3]. In the first stage, some constraints are relaxed, which are then recovered in the second stage. They show that this approach saves computation time, while maintaining the solution quality.

4 Approach

To achieve the goal of this research project, soft constraints need to be defined by the intended audience. Hence, the first step is to collect feedback from students and staff regarding scheduling preferences. These preferences are then used to define constraints. A schedule tailored to a specific audience is usually in turn best evaluated by the very same audience. This is however an arduous task, as it would require qualitative testing with uncountably many external variables. Hence, evaluation metrics need to be set up so that timetables can be compared and improved, and so that the performance between different algorithms and heuristics become clear.

Regarding algorithms and heuristics, a step-by-step approach will be used. Initially, only the hard constraints will be used together with a baseline greedy algorithm. Following this, soft constraints will be added to the program one by one. This allows different combinations of soft constraints to be compared. Finally, we hope to investigate the performance of ILPs and metaheuristics, such as evolutionary algorithms, on the feasibility of the schedules. We plan to proceed with the project as follows:

1. Collect feedback from students and staff regarding scheduling preferences.
2. Define constraints from these preferences.
3. Derive evaluation metrics.
4. Develop a generic framework which allows us import, export and evaluate schedules.
5. Set up a greedy algorithm to build a schedule, using only hard constraints.
6. Add soft constraints.
7. Investigate the performance of ILPs and metaheuristics, such as evolutionary algorithms.

5 Deliverables

During the first phase, a presentation will be given about the current status of the project, the state of the art and our research into timetable scheduling. Said presentation complements this very project plan. Furthermore, together with the presentation in the first phase, we will submit a draft of the questionnaire which will help us to adjust weights for the constraints.

By the end of the second phase, we should have a sufficient collection of preferences submitted by students and lecturers such that we can evaluate the outcomes and finalize an evaluation metric on the schedules produced by the algorithms. In order to evaluate the schedules, we want to have a generic framework which allows us to import and build schedules with various algorithms as mentioned in Section 4. This framework will be submitted along with the datasets used to test the basics of a greedy algorithm, an ILP approach and metaheuristic algorithm(s). The outcomes of initial tests will be discussed in a presentation together with a layman's report.

In the third and final phase of this project, at least the three aforementioned algorithms will all be fully functional. Further testing on the produced schedules shall be conducted and evaluated. A final report with all findings and an extensive presentation on the whole project will be submitted at the end of this phase. Optionally, a user interface may be built around the algorithms to allow the usage of these by staff members at DKE who currently develop the department's schedules.

6 Time Management

As mentioned in Section 5, this research project consists of three phases. The first one has already been accomplished by handing in this project plan for the next two phases. This section provides details regarding the time management plan that will be followed during the course of the project and the relations between each task, indicating if some of them can be done concurrently.

The selected management methodology is Scrum with three sprints, all three lasting two weeks each. As described in the Gantt diagram (Figure 1), before the first sprint, one week is reserved for research, for creating and setting up the framework, and for gathering requirements for the future implementations. Then, the first sprint (20/04/2020 to 05/05/2020) consists of data gathering for future validation of the timetables together with the implementation of the greedy approach, and possibly the implementation of another approach if time allows This sprint ends with the deployment of the first alpha version of the scheduling software.

The second sprint (06/05/2020 to 05/06/2020) focuses on further and more complex approaches for the scheduling system, such as ILP and metaheuristics. After testing and client review, the beta version is ready for deployment. The second week of this sprint marks the start of the report writing.

The last sprint corresponds to the whole of Phase 3 (15/06/2020 to 02/07/2020). This means that this sprint is three weeks long instead of two. It is the most important sprint, since four milestones need to be accomplished: deploying the 1.0 version of the scheduling software, delivering final software to client after validation, delivering the final report and giving the final presentation.

The Scrum master will switch in every review meeting so that every member involved will be the coordinator at some time during the project. Furthermore, if we come up with a new idea or possible implementation for the framework during the implementation process, it will be considered during the next sprint planning.

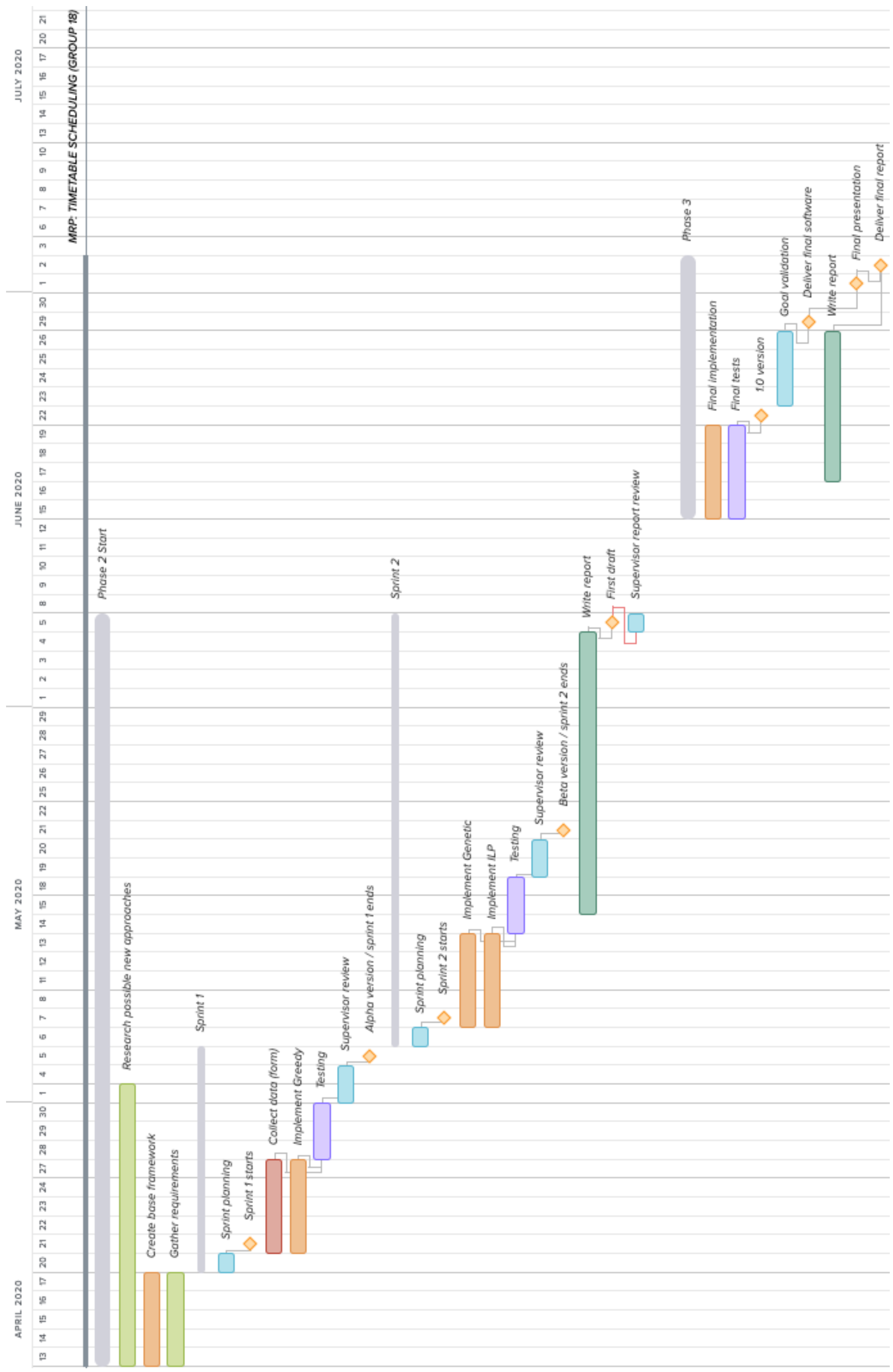

Figure 1: Gantt diagram with the project organization. Light green bars represent research, orange is implementation, blue corresponds to organization, red is data gathering, magenta represents a testing task and dark green stands for report writing.

7 Risk Analysis and Contingency Plans

Timetable generation is a challenging task and typically requires a lot of time since it is an NP-hard problem [2]. There are several factors which may have an impact on the success of this project. In this section, we will analyze these risks and describe possible solutions which minimize their effects.

A major risk that we face is that the current timetables cannot be improved upon by the implemented algorithms despite the added heuristics, at least from a student's perspective. This would mean that the timetables created by the algorithms satisfy less or the same number of constraints than those that are currently being used. In the case that the generated schedules are similar to the current schedules, a beneficial outcome would be that the schedules can now be generated automatically rather than be made completely by hand.

A second problem is that satisfying all soft constraints, these being the wishes of the teaching staff and the student body, may prove to be too difficult a task. Some constraints are more important than others; the hard constraints, set up to prevent problems like double-booked locations, should be taken into account with the highest priority. It is only when the hard constraints have been satisfied that the soft constraints can be considered.

Last but not least, due to the outbreak of COVID-19, most physical activities and face-to-face meetings will not happen as planned. Therefore, weekly group meetings will be performed online. Furthermore, the delayed exams of period four are currently planned to take place in June, which conflicts with the deadline of this project. As a result, we likely have less time to work on the project. To limit this risk, we have planned in the Gantt chart from Figure 1 that the algorithms should be working in phase two and that we begin to write a draft of the final report. Lastly, it is possible that one or more members of the project group are affected by the COVID-19 outbreak for an extended time period and cannot contribute as much as planned. In this case, the other members will collectively take on the tasks assigned to those affected for as long as this may be required.

References

[1] Victor A. Bardadym. Computer-aided school and university timetabling: The new wave. In Edmund Burke and Peter Ross, editors, *Practice and Theory of Automated Timetabling*, pages 22–45. Springer Berlin Heidelberg, 1996.

[2] Tim B Cooper and Jeffrey H Kingston. The complexity of timetable construction problems. In *International Conference on the Practice and Theory of Automated Timetabling*, pages 281–295. Springer, 1995.

[3] Sophia Daskalaki and Theodore Birbas. Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1):106–120, 2005.

[4] Sophia Daskalaki, Theodore Birbas, and Efthymios Housos. An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research*, 153(1):117–135, 2004.

[5] Itc 2019: International timetabling competition. https://www.itc2019.org/home.

[6] Itc 2007: Finalist ordering. http://www.cs.qub.ac.uk/itc2007/winner/finalorder.htm.

[7] Rhydian Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR spectrum*, 30(1):167–190, 2008.

[8] International series of conferences on the practice and theory of automated timetabling (patat). https://patatconference.org/.

[9] Sanja Petrovic and Emilio Burke. University timetabling. 01 2004.

[10] Comprehensive university timetabling system (unitime). https://unitime.org/.