



University timetable scheduling

Bachelor Project Report

Aron Fiechter

2018

Advisors: Michele Lanza, Andrea Mocci, Luca Ponzanelli

Abstract

- ▷ Context: Creating and managing timetables of courses is an issue for many institutions because of various constraints that need to be respected in the planning.
- ▷ Problem: This is in fact also a problem for the dean's office of Faculty of Informatics at USI, which currently creates timetables manually.
- ▷ Solution: As a solution, this process can be automated using state of the art tools to solve planning problems.
- ▷ Teaser: In this bachelor project I [we? passive?] design a web application (a module of the MyUSI platform) that allows to automatically create timetables given a set of constraints, and also to customize proposed solutions and visualize personalised schedules.

1 Introduction

1.1 Motivation

Creating a timetable is an issue for many institutions because of the various constraints that might need to be respected, such as different room sizes, instructor availability, elective courses, different frequency of courses. For example, this is a problem for the Faculty of Informatics at USI as well: the Atelier courses need to be placed in the afternoons, there are three rooms that can contain 60 people and four that can contain 30 people, there are different elective courses, and not all professors are available everyday.

1.2 Goal

The goal of this project is to create a web application (a module of the MyUSI platform) that allows to automatically create timetables given a set of constraints, and also customize proposed solutions; the resulting calendars should also be available to single students in the form of semester schedules, or personalised schedules in the case of elective courses and courses from different years.

Additional Goals

Once the basic system is implemented, possible extensions include managing timetables of the other faculties, adding single non-lecture events to sched-

ules (such as seminars, talks), visualising statistical information about room usage.

1.3 Technologies

The web application will use Scala on the server side, OptaPlanner¹ to solve the optimization problem, and Polymer 2.4 on the client side.

2 State of the art

2.1 Planning problem

2.2 OptaPlanner

3 Approach

3.1 Domain model

3.2 Constraints

3.3 Interface

3.4 Web application UI

4 Evaluation

4.1 Case study on real data

4.2 Examples results

5 Conclusion

5.1 Foo

6 Bibliography

7 Appendix

¹<https://www.optaplanner.org/>