Noname manuscript No.
(will be inserted by the editor)

Modeling and Methods in Untis, a Popular Software System for School Timetabling

Sebastian Knopp

Received: date / Accepted: date

Abstract Untis is a company focusing on school timetabling since 1970. Its software is used by more than 25 000 educational institutions worldwide and satisfies the needs of a diverse range of users (including elementary and secondary schools, vocational schools, or universities of applied sciences). The majority of the customers apply the integrated optimization solver. Enabling each school to solve its individual timetabling problem requires taking into account a large range of constraints that are arbitrarily combinable. Therefore, a quite general optimization model is necessary. The modeling in the Untis software includes all 28 constraints listed in the survey of [Pillay(2014)] plus further important aspects such as the consideration of multiple time grids or an integrated planning over a whole year. The modelling incorporates only few hard constraints and more than fifty soft constraints. The optimization uses sophisticated construction and improvement heuristics which are embedded in different meta-heuristic approaches. While optimizing in a batch-mode (see [Schaerf(1999)]) is the core feature of the Untis software, various interactive modes, supported by the underlying optimization model, are offered to ease the work of the timetabler.

We address three topics in this talk. First, we outline core modelling aspects of the main timetabling problem addressed in the software. Second, we discuss the integrated planning over a whole school year, where timetables might change from week to week. Finally, some features of the interactive planning modes are shown.

First, let us outline the modeling and its core assumptions. We consider a week divided into a grid of periods, specifying times during which lessons can take place. The main resources involved are classes, teachers, and rooms. The goal is to schedule a given a set of lessons. Each lesson is assigned to a subject, such as music or mathematics. Each lesson must take place for a given number

Sebastian Knopp
Untis GmbH, A-2000 Stockerau, Belvederegasse 11
E-mail: sebastian.knopp@untis.at

of times during the week and requires a given set of resources. Additionally, we are given a set of couplings. A coupling is a set of lessons which must take place simultaneously. Coupling constraints are always observed. This is ensured by the solution representation. All other constraints are included as soft constraints which cover a broad range of requirements, including didactic aspects, workload constraints, resource utilization constraints, and time preferences. Violations of soft constraints are rated using penalty scores. The objective function is to minimize a total weighted penalty score.

Second, we survey the temporal planning over the course of a year. A feature important for many schools which has found only little attention in the literature. Often, timetables can differ from week to week. For example, some lessons might take place only every second week instead of every week. Other lessons should be taught only once a year during a block of three consecutive weeks, which is a common case for vocational schools. Also, differences in the curriculum between the first and the second half-year must often be taken into account. To address such requirements in the problem formulation, we assign to each lesson a set of calendar days during which the lesson is allowed to take place. Additionally, consistency is required as follows: The same lesson must take place at the same times during all weeks (if it takes place). In combination with coupling constraints, this modeling allows, e.g., to enforce two weekly alternating subjects which take place during the same period every week. These constraints are included in the optimization algorithm in an integrated way.

Finally, we show how the modeling as an optimization problem is applied to assist the user with the creation of a timetable. Usually, the timetabling process starts with importing or entering data and constraints. After that, a first and quick optimization run allows the user to assess if major problems exist in the input data. Problems might be caused by an extensive use of coupling constraints or a combination of constraints which is too rigid. The user can spot such issues by using an analysis tool which searches for maximum cliques in a conflict graph. Once all the inputs are completed, the user can start the thorough optimization run which creates a high-quality timetable. After that, manual modifications to the timetable can be made. Their impact on the objective function is immediately visualized in the user interface. In case a user dislikes the placement of a particular lesson, the software can suggest complex moves which shift that lesson. The objective function is taken into account to find suitable displacements for the lessons involved.

Keywords School Timetabling · Heuristics · System Demonstration

References

[Pillay(2014)] Pillay N (2014) A survey of school timetabling research. Annals of Operations Research 218(1):261–293

[Schaerf(1999)] Schaerf A (1999) A survey of automated timetabling. Artificial intelligence review 13(2):87–127