Examination timetabling with Moses: System demonstration

An IT system for academic resource planning at universities

Gerald Lach · Mirjana Lach · Julian Steenken · Erhard Zorn

Keywords Examination timetabling · Automated timetabling · Integer programming · Resource planning · Higher education

1 Introduction

Every semester, universities face the complex problem of planning the examination period(s). Usually, there are two such periods—the first one is immediately after the lecture period, while the second one is before the next lecture period starts.

As one of the consequences of the Bologna Process, the number of written examinations per student at European universities has increased. Consequently, students face an increased strain in completing their studies within the prescribed program length.

Obviously, the university has to offer an examination timetable that permits students who attend courses (and examinations) according to their reg-

Gerald Lach
Technische Universität Berlin
Institute of Mathematics/innoCampus and
MathPlan GmbH
E-mail: gerald.lach@mathplan.de

Mirjana Lach
Technische Universität Berlin
Institute of Mathematics/innoCampus
E-mail: mlach@math.tu-berlin.de

Julian Steenken
Technische Universität Berlin
Institute of Mathematics/innoCampus
E-mail: steenken@math.tu-berlin.de

Erhard Zorn
Technische Universität Berlin
Institute of Mathematics/innoCampus
E-mail: erhard@math.tu-berlin.de

ular study plan to take all written exams without facing time conflicts. In addition, it is desirable that students do not have more than one exam per day. Furthermore, students appreciate a time interval between examinations—e. g. a day for final revisions.

Hence, the necessity of automated examination timetabling is stronger than ever. Particularly at large institutions like Technische Universtät Berlin (TU Berlin), which has more than 32,000 students, the examination timetabling problem is highly complex. This is due to not only the size of the university but also its dynamically changing framework—e. g. curricular alterations, changing teaching staff, availability of rooms, and so on.

Since 2002, the IT system *Moses* has been developed at *innoCampus*, a department of TU Berlin, in order to solve university timetabling problems. In addition to solving course timetabling and post-enrollment-based timetabling problems, *Moses* computes examination timetables that satisfy a wide range of soft and hard constraints.

In spring 2003, *Moses* was used for the first time at TU Berlin to distribute students to their tutorials (additional small exercise classes offered for courses with large numbers of students) in mathematics courses. Today, more than 80 large courses distribute their students into more than 1,000 tutorials (for details see [1]).

Due to its great success, *Moses* has been extended to solve the examination timetabling problem and the university course timetabling problem as well. Since 2013, *Moses* has been used at RWTH Aachen University (around 44,000 students) to solve the university course timetabling problem. One-and-a-half years later, TU Berlin also stopped copying the old course timetables and started to create them automatically using Moses [3].

Due to the high quality of the *Moses* examination timetabling results at TU Berlin, the Technical University of Munich (TU München), which has more than 39,000 students, asked to use *Moses*. In 2015, a TU Berlin spin-off deployed *Moses* at TU München. Since then, large-scale examination timetabling problems involving more than 1,100 written examinations each semester have been solved.

2 Approach

Our approach to solving examination timetabling problems even at large universities is based on integrating powerful mathematical optimization algorithms with efficiently designed workflows supported by an IT system. We found that for a successful performance of an automated timetabling system, the workflow is as important as powerful optimization tools.

Since the results of our optimization strongly depend on the quality of input data, we designed a well-structured user interface in order to ensure high usability.

3 Process

Two distinct user roles are involved in a timetabling process managed by *Moses*: On the one hand, there are the users—e. g. lecturers. On the other hand, there are the supervisors managing the timetabling process. The work flow for the course timetabling is separated into six subsequent steps:

1. Preparation (supervisors): updating facility database among other things
2. Data input (users): inserting data through the *Moses* web application—e. g. desired dates, room(s), and anticipated number of participants
3. Data cleaning (supervisors)
4. Timetable creation (supervisors) using optimization tools
5. Internal publication of the timetable (supervisors) for revision (users)
6. Publication of the timetable (supervisors)

4 System Architecture

The system comes as a *Java Enterprise Edition* web application running on the latest version of the *Glassfish Application Server*. The user interface is based on *Java Server Faces (JSF)* and the comprehensive JSF-based *Primefaces* library. Beyond that, modern responsive design (mainly based on the front-end framework *Twitter Bootstrap*) enables users to access the system from a device of their choice.

The optimization consumes a lot of processing power and, therefore, is computed on a separate machine than the one on which the application server is installed. A dedicated distribution server communicates with the application server and controls the optimization processes. This distribution server has access to a certain number of optimization machines to which the scheduling tasks are assigned.

Typically, the IT landscape of a university—consisting of different systems and platforms—differs considerably among the different institutions. That is why the *Moses* system offers various interfaces to other systems for importing and exporting data automatically, semi-automatically, or manually.

5 Optimization

As described in [2], we developed a solution method for the examination timetabling problem using linear integer programming as applied in [4].

The solution procedure involves two steps: First, time slots are assigned to examinations so that there are no time conflicts within any given course of studies. In a second step, rooms are assigned to the scheduled exams. Both steps can be executed independently without loss of optimality because only schedules that allow for a conflict-free room assignment are considered feasible in the first step.

For details, please refer to the afore-mentioned papers.

6 Future Work

The bigger the system grew, the more obvious the need for modularization became. Decomposition into pluggable modules, therefore, is the most prioritized development task. Due to the extensive and heavily cross-linked database and the interconnected general structure, this is expected to be challenging. A strict modularization would extend the systems adaptability to the specific needs of the universities employing it. For the same reason, internationalization and pluggable branding are other goals worth pursuing.

References

1. Höner, J., Lach, G., Zorn, E.: An IP-based model for the post-enrollment-based course timetabling problem at TU Berlin. In: Z. Hanzálek, G. Kendell, B. McCollum, P. Šůcha (eds.) MISTA 2015. Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications, pp. 331–344. Prague, Czech Republic (2015)
2. Lach, G., Lach, M., Zorn, E.: Examination timetabling at Technische Universität Berlin. In: Z. Hanzálek, G. Kendell, B. McCollum, P. Šůcha (eds.) MISTA 2015. Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications, pp. 260–266. Prague, Czech Republic (2015)
3. Lach, G., Lach, M., Zorn, E.: Solving huge real-world timetabling instances. In: Z. Hanzálek, G. Kendell, B. McCollum, P. Šůcha (eds.) MISTA 2015. Proceedings of the 7th Multidisciplinary International Conference on Scheduling: Theory and Applications, pp. 370–378. Prague, Czech Republic (2015)
4. Lach, G., Lübbecke, M.: Optimal university course timetables and the partial transversal polytope. In: C. McGeoch (ed.) 7th International Workshop on Efficient and Experimental Algorithms (WEA08), *LNCS*, vol. 5038, pp. 235–248. Springer, Berlin (2008)