Comprehensive University Timetabling System

http://www.unitime.org
Introducing UniTime

What is Educational Timetabling?
• The process of assigning classes (or exams) to times and locations
• A difficult optimization problem with many competing objectives: student conflicts, faculty requirements, space constraints

Why is it needed?
• Minimize student conflicts, thus help students receive degrees on time
• Help use resources effectively
• Make process easier to manage (knowledge transfer and cooperation)
• Fairness and satisfaction with the timetable
• What-if scenarios
• Ability to adapt to changes

What is UniTime?
• Comprehensive academic scheduling solution
• Four components: course timetabling, examination timetabling, student scheduling, event management
• Open Source (Apache License, Version 2.0)
• Web-based, written in Java using modern technologies
• Using state-of-the-art optimization algorithms
• Distributed data entry and timetabling in multi-user environments
• Easy to extend and/or customize
• Has been applied at large institutions (up to 40,000 students)
• Sponsored project of the Apereo Foundation
• Over 500k lines of code (including the constraint solver)
• Used by institutions around the world
  - USA, Czech Republic, Pakistan, Croatia, Poland, Turkey, Peru, Kuwait, …

For more information…
• Web site: http://www.unitime.org
• Online demo: https://demo.unitime.org
• Source codes: https://github.com/UniTime
• Email: support@unitime.org
Course Timetabling

Lifecycle of a course timetable

Goal
Assign class times and locations such that:
- All hard constraints and other requirements are met
- Desirable objectives are satisfied as much as possible
  - Minimize student conflicts
  - Accommodate time and room preferences
  - Allow preferred class time distributions
  - Fairness, minimize travel times, etc.

Constraints
- Room sizes, equipment, and availability
- Faculty time, room requirements and preferences
- Structures of courses that are to be offered
- Student course demands (curricula, pre-registration, etc.)

Student Conflicts
A student cannot take a combination of courses
1. Classes overlap in time
   *(or one after the other in rooms that are too far apart)*
2. There is not enough space in a non-overlapping combination of classes
Course Structure

Classes are organized by the course structure
• Intuitive data entry and display of classes and their requirements
• Helps to define how students can enroll into the course
• Additional relations can be derived from the structure

<table>
<thead>
<tr>
<th>MA 170</th>
<th>STAT 170</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>Laboratory</td>
</tr>
<tr>
<td>40 Full Term</td>
<td>40 Full Term</td>
</tr>
<tr>
<td>50 1 x 50</td>
<td>150 3 x 50</td>
</tr>
<tr>
<td>Classroom</td>
<td>EDUC CompPr</td>
</tr>
<tr>
<td>G. Newman</td>
<td>J. Smith</td>
</tr>
</tbody>
</table>

Example of a course structure

Constraint-based Solver
• Can be used in modes between manual and fully automated
• Local search based framework using constraint programming primitives
• Winner of two tracks of the International Timetabling Competition 2007 (finalist of all three tracks)
• Applicable to a variety of constraint satisfaction and optimization problems

More features
• Course management
• Data exchange / roll-forward
• Room distances and travel times
• Date patterns
• Clustering
• …
Student Scheduling

**Goal:** Enroll students to classes in a way that maximizes the ability of students to get the courses they need

- Students fills in course requests
- System suggests a schedule that best meets student needs
- Students can make later modifications to schedule

**Why is it needed?**
To ensure that students will be able to get the courses they need when multiple sections are offered.

Batch Scheduling

- All students are scheduled at one time after the timetable is produced based on student pre-registrations
- An optimization process, using the student scheduling solver

Online Scheduling

- Students without pre-registrations can enroll online (e.g., incoming freshmen)
- All students can make adjustments to their schedules
- Automatically hold space in sections based on historical student demand
- Reservations, automated wait-lists, instructor consents, advisor roles, ...
Examination Timetabling

**Goal:** Assign examinations to time periods and rooms

- An exam can be offered for a class, a course, or a combination of these
- No two exams in the same period and room
- Examinations must fit the period and room
- Room must be available and all period, room, and distribution requirements must be met
- Desirable objectives are to be satisfied as much as possible
  - Minimize student conflicts (direct, back-to-back, more than two exams on a day)
  - Period, room, and distribution preferences
  - Minimize room splits, distance to original room, large exams first, rotation, ...
  - Student and instructor availability is considered

---

**Event Management**

Management of the remaining classroom space

- Fully distributed, including an approval process
- Students and instructors can see their classes, examinations, course-related and other events through the event management; they can also request events for the event manager to approve
- Export to CSV, PDF, iCalendar, JSON with the ability to subscribe to a particular schedule