UNI|Time: University Course Timetabling & Student Scheduling System

System Demonstration

Tomáš Müller
Keith Murray
Stephanie Schluttenhofer

ICAPS 2007
UNI|Time: Motivation

Course Timetabling & Student Sectioning Problem at Purdue University

- Create and modify course timetables that better meet student course demands
- And allow students to be assigned to the courses in a way that minimizes conflicts
- Large scale university-wide problem
  - 9,000 classes, 570 rooms, 39,000 students with 259,000 class requests
- Allow decomposition to several problems (large lectures, departmental timetables)
- Departmental schedule managers responsible for their own solutions
UNI\textit{time}: System Architecture

- **Presentation Layer**
  - Solver Operation & Timetable Presentation
  - Load Data
  - Save Solution

- **Timetabling Layer**
  - Timetabling specific variables, values, constraints, heuristics related to V,D,C,f

- **Constraint Satisfaction Layer**

- **User Interface**
  - Business Logic
  - Course Structure Model
  - Timetable Solutions
  - Database

- **Solver Implementation**
  - Problem Specific Heuristics
  - Timetabling & Student Sectioning
  - Solver Data Model

- **Abstract CSOP (V,D,C,f) Solver**
  - General Variable and Value Selection Heuristics
UNI|Time: System Architecture

- **Web Server**
  - Server-client application with web-based interface
  - Written in Java, using J2EE, Hibernate, and SQL-enabled database
  - Supports coordinated work on timetable in a multi-user environment

- **Solver**
  - Based on Iterative Forward Search (IFS) algorithm
    - A mixture of local search and backtracking
    - Gradually extends (partial) feasible assignment
    - Applicable to various problems and scenarios
    - Easily extensible
  - Problem model and constraints consider complexity of all university courses
    - Interaction between problems
    - Competitive behavior (fairness of the solutions among departments)
  - Data consistency
    - Ability to identify and present to the user any inconsistencies and potential problems in the input data
UNI|Time: Course Timetabling

- For each class
  - Student requirements
  - Time requirements & preferences
    - Meeting patterns
e.g., 3 x 50 min, 2 x 75 min
  - Room requirements & preferences
    - Capacity
    - Required equipment
    - Room / building preference
    - Building distances
  - Instructor
  - Additional (distribution) constraints
    - Between several classes
  e.g., back-to-back, precedence
  - Other
    - Departmental balancing,
efficient utilization of time and rooms, …
UNI|Time: Data Management

- Data Management (instructional offering structure)
  - Classes are organized in a visual representation of the course structure
    - GUI allows intuitive entry and display of class and constraint data
    - Preferences and requirements can be set at multiple levels
    - Some constraints are automatically deduced from course structure

--- Preferences ---

<table>
<thead>
<tr>
<th>Demand</th>
<th>Mins Per Week</th>
<th>Limit</th>
<th>Time Pattern</th>
<th>Time</th>
<th>Room</th>
<th>Distribution</th>
<th>Instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA 170</td>
<td>62</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAT 170</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>40</td>
<td>1 x 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laboratory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>40</td>
<td>3 x 50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lec 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>40</td>
<td>1 x 50</td>
<td></td>
<td></td>
<td></td>
<td>S. Bell</td>
</tr>
<tr>
<td>Lab 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>20</td>
<td>3 x 50</td>
<td></td>
<td></td>
<td></td>
<td>J. Beckley</td>
</tr>
<tr>
<td>Lab 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>20</td>
<td>3 x 50</td>
<td></td>
<td></td>
<td></td>
<td>J. Beckley</td>
</tr>
</tbody>
</table>
UNI|Time: Modifying Solutions

- Using Automated Solver: Minimal Perturbation Problem
  - Solution to a modified problem is as close as possible to the initial solution
- Manually: Interactive Mode
  - Solver is guided by the user, providing an evaluated list of choices
  - Backtracking with limited depth is used

<table>
<thead>
<tr>
<th>Score</th>
<th>Class</th>
<th>Date</th>
<th>Time</th>
<th>Room</th>
<th>Students</th>
</tr>
</thead>
<tbody>
<tr>
<td>+15.2</td>
<td>POL 101 Lec 3</td>
<td>Full</td>
<td>TTh 12:00p → TTh 7:30a</td>
<td>BRNG 2280</td>
<td>+11</td>
</tr>
<tr>
<td>+31.7</td>
<td>POL 101 Lec 3</td>
<td>Full</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+36 (h+3)</td>
</tr>
<tr>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full</td>
<td>TTh 10:30a → TTh 1:30p</td>
<td>BRNG 2280 → BRNG 2290</td>
<td></td>
</tr>
<tr>
<td>+36.6</td>
<td>POL 101 Lec 3</td>
<td>Full</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+36 (h+4)</td>
</tr>
<tr>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full</td>
<td>TTh 10:30a → TTh 7:30a</td>
<td>BRNG 2280</td>
<td></td>
</tr>
<tr>
<td>+44.1</td>
<td>POL 101 Lec 3</td>
<td>Full</td>
<td>TTh 12:00p → TTh 10:30a</td>
<td>BRNG 2280</td>
<td>+34 (h+2)</td>
</tr>
<tr>
<td></td>
<td>HIST 342 Lec 1</td>
<td>Full</td>
<td>TTh 10:30a → TTh 3:00p</td>
<td>BRNG 2280 → BRNG 2290</td>
<td></td>
</tr>
<tr>
<td></td>
<td>OBIR 330 Lec 4</td>
<td>Full</td>
<td>TTh 3:00p</td>
<td>BRNG 2290 → LWSN B155</td>
<td></td>
</tr>
</tbody>
</table>

(All 1571 possibilities up to 3 changes were considered, top 4 of 17 suggestions displayed)

Ability to incorporate changes into an existing solution is critical in real-life problems
UNI|Time: Student Sectioning

- Student requests courses, system determines classes (sections)
  - Respects course structure, reservations, and student preferences
UNI|Time: Student Sectioning

- **Initial Sectioning (during timetabling)**
  - Pre-registration, last like data for first year students, projected changes
  - Timetabling solver minimizes potential student conflicts

- **Final Sectioning**
  - Once the timetable for the whole university is created
  - Registration of classes for students, reservations, wait lists

- **Online Sectioning**
  - Registration of first year students and other late registrants
  - Changes in existing enrollments
  - Expected students demands are used to direct students from sections with excess demand
  - Computed in final sectioning, updated with each new student
UNI|Time: Web Site

- URL: http://www.unitime.org
  - Available for download:
    - Course Timetabling & Student Sectioning application described here
      - Open Source (GNU GPL)
    - Constraint Solver library
      - Including Course Timetabling and Student Sectioning extensions
        - Open Source (GNU LGPL)
  - Online documentation
  - Ongoing research
    - Publications & presentations
  - Benchmark data sets
    - Real-life data for course timetabling and student sectioning problems
  - Contact: research@unitime.org