Purdue University Timetabling

- University-wide problem size
  - 9,000 classes, 570 rooms
  - 39,000 students with 259,000 class requests

- Problem Decomposition
  - Central timetable for large lecture classes
    - Approximately 900 classes, 54 rooms
    - Utilization over 78% (~ 97% for four largest rooms)
  - Timetables for individual departments
    - 70 timetables with sizes from 10 to 750 classes
    - Built on top of large lecture timetable
    - Departmental schedule managers are responsible for their own solutions
  - Central computer laboratory timetable
Purdue University 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, …

Each student states which courses he or she wants to attend (soft constraint)
Purdue University 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, …
Purdue University 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, …
Purdue University Timetabling

- **User Interface**
  - Server-client application with web-based interface
  - Written in Java, using J2EE, Hibernate, and Oracle Database
  - Supports coordinated work on timetabling in a multi-user environment

- **Solver**
  - Iterative Forward Search (IFS) algorithm
    - A mixture of local search and backtracking
    - Works in iterations
    - Gradually extends (partial) feasible assignment
    - Applicable to various problems and scenarios
  - Problem model and constraints consider complexity of all university courses
Critical Aspects of Application

- Interaction between problems
  - Only committed solutions are visible and considered by other problems
    - Consistency is ensured between committed solutions
    - Room sharing
      - At any time, a room is either unavailable, available for use on a first come (commit) first served bases, or allocated to a particular department
    - Mutual constraints (e.g., student enrollments) are considered only between the current problem and solutions to committed problems
  - If there are many relations between two (or more) departments
    - E.g., many students are taking classes from both departments
    - These departments can be solved together
      - A timetable containing all classes of these departments is created
    - Or agree on a solution order
      - E.g., the more difficult problem can be solved and committed, the second timetable is built on top of the first.
Critical Aspects of Application

- 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 the structure

<table>
<thead>
<tr>
<th></th>
<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></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STAT 170</td>
<td>62</td>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lecture</td>
<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>150</td>
<td>40</td>
<td>3 x 50</td>
<td></td>
<td></td>
<td></td>
<td>ENAD</td>
<td>BTB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dell 2.8 machines</td>
<td></td>
</tr>
<tr>
<td>Lec 1</td>
<td></td>
<td>50</td>
<td>40</td>
<td>1 x 50</td>
<td></td>
<td></td>
<td>Classroom</td>
<td>S. Bell</td>
</tr>
<tr>
<td>Lab 1</td>
<td></td>
<td>150</td>
<td>20</td>
<td>3 x 50</td>
<td></td>
<td></td>
<td>ENAD</td>
<td>BTB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dell 2.8 machines</td>
<td>J. Beckley</td>
</tr>
<tr>
<td>Lab 2</td>
<td></td>
<td>150</td>
<td>20</td>
<td>3 x 50</td>
<td></td>
<td></td>
<td>ENAD</td>
<td>BTB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Dell 2.8 machines</td>
<td>J. Beckley</td>
</tr>
</tbody>
</table>
Critical Aspects of Application

- Competitive Behavior (fairness of the solution)
  - Preferred times and rooms
    - Minimization of the overall cost (objective function) typically favors those who provide the most preferences
  - Normalization of time preferences
    - Increasing the number of preferences lowers individual preference weights
  - Departmental balancing constraint
    - Classes from a department are evenly spread across available times
Critical Aspects of Application

- Data Consistency Checking
  - Ability to find a solution
    - Input data often contain inconsistencies preventing a complete solution from being found
    - Therefore, the first stage of the timetabling process is to verify data and identify the weaknesses
  - Providing feedback to the user
    - Solver must be able to provide information in an easily readable form
- Conflict-based statistics identify problem areas
Critical Aspects of Application

- Interactive Changes (ability to alter a solution)
  - Solutions can be manipulated manually or by fully automated solver
  - Ability to incorporate changes into an existing solution is critical in real-life problems
    - 1) Minimal Perturbation Problem
      - Solution to a modified problem is as close as possible to the initial solution
    - 2) 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>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PHIL 330 Lec 1</td>
<td>08/21-12/17</td>
<td>MWF 4:30p</td>
<td>CL50 224 → WTHR 200</td>
</tr>
<tr>
<td>0.8</td>
<td>PSY 120 Lec 4</td>
<td>08/21-12/17</td>
<td>MWF 4:30p</td>
<td>WTHR 200 → CL50 224</td>
</tr>
<tr>
<td>5.75</td>
<td>PHIL 330 Lec 1</td>
<td>08/21-12/17</td>
<td>MWF 4:30p</td>
<td>CL50 224 → LILY 1105</td>
</tr>
</tbody>
</table>
Critical Aspects of Application

- **Student Sectioning**
  - Student requests courses, system determines classes (sections)
  - Student Enrollments (for timetabling)
    - Pre-registration, last like data for first year students, projected changes
    - Solution is created based on these data

- **Work in progress**
  - Final Student Sectioning
    - Registration of classes for students, reservations, wait lists
  - Online Student Sectioning
    - Precompute expected conflicts based on final sectioning
    - Registration of first year students and other late registrants
    - Changes in existing enrollments
Demonstration