What is Academic Scheduling?

Academic scheduling is the assignment of times, rooms, instructors, and students to classes. There are two traditional approaches.

Master Scheduling (current process):
• Develop class timetable
• Schedule students to classes in timetable

Demand-driven Scheduling:
• Collect student demand for courses and times
• Develop optimized timetable and student schedules
Purdue University has come to rely on efficiencies resulting from optimized scheduling.

- Cost of offering classes
- Limited classroom space

Academic Scheduling functionality is not included in any ERP packages.

Timetabling and Scheduling are active research areas with very promising results for improving University efficiency and meeting student needs.
Need for Timetabling at Purdue

1. Demand by schedule deputies for help creating departmental class schedules

2. Scarcity of lecture rooms making large lecture timetable too difficult to create manually

3. Need for tool to modify existing timetable (class limit changes, ADA accommodations, etc.)

4. No additional resources to support enrollments ⇒ need for increased efficiency
Need for Student Sectioning

Student conflicts caused by individual class time choices

e.g., students A and B each require courses 1 and 2, section a of each course meets at same time:
Need for Student Sectioning: Experiments

Seniority Order Registration with Time Selection

<table>
<thead>
<tr>
<th># of Classes not Scheduled</th>
<th>Students w/ Unscheduled Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2693</td>
</tr>
<tr>
<td>2</td>
<td>513</td>
</tr>
<tr>
<td>3</td>
<td>94</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3316</td>
</tr>
</tbody>
</table>

Fall ‘04 Requests 38429

Incomplete Schedules in Scheduling Sequence

Students Processed

Incomplete Schedules

0 1000 2000 3000 4000 5000 6000 7000 8000 9000 10000

0 500 1000 1500 2000 2500 3000 3500 4000 4500 5000
Create and modify class timetables that better meet student course demand and allow students to be assigned to sections that minimize conflicts.

Improve ability to meet student class time needs/preferences.
Current Research and Related Work

Purdue timetabling research began 6 years ago
• Collaboration with Masaryk and Charles Universities
• Extensive knowledge of scheduling and constraint-based optimization
• Published work has been well-received by research community
• Have a very robust and scalable constraint-based solver framework

Constraint Programming Techniques
• Powerful tool for solving optimization problems
• Problem is described in natural way (variables, values, constraints)
• Practical applications in planning, timetabling, scheduling

Other University Work
• ASAP group, U. of Nottingham, strong research base ⇒ exam system
• MIT GASP project, develop student sectioning and timetabling
• Many presented projects (usually not at university-wide level)
Constraint Satisfaction Problem (CSP)

Problem $\Theta = (V, D, C)$
- $V = \{v_1, v_2, \ldots, v_n\}$ is a finite set of variables
- $D = \{Dv_1, Dv_2, \ldots, Dv_n\}$ is a set of domains
  - Domain is a finite set of values
- $C = \{c_1, c_2, \ldots, c_m\}$ is a set of constraints
  - A constraint limits the combination of values that variables can simultaneously take

Solution
- Assignment of variables
  - $\eta \subseteq \{ v/a \mid v \in V \land a \in Dv \}$
  - where $\forall v/a, w/b \in \eta \quad v = w \Rightarrow a = b$
  - That is complete $|\eta| = |V|$
  - That satisfies all constraints from $C$

Over-constrained problems:
Looking for a maximal assignment that satisfies all constraints from $C$
Problem $\Phi = (V, D, C, f)$

- $(V,D,C)$ is a constraint satisfaction problem (CSP)
- $f$ is an objective function
  - That maps every partial feasible assignment to a number
    - feasible ~ all hard constraints are satisfied
    - Usually expressed by soft constraints
      - $f(\eta) = \text{(weighted) sum of soft constraints that are not satisfied}$

Solution

- Complete assignment of variables $\eta$
  - that satisfy all the constraints from $C$
  - so that $f(\eta)$ is minimal (or maximal)
# Course Timetabling Model

<table>
<thead>
<tr>
<th>Variables:</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domains:</td>
<td>Values of time and room assignments</td>
</tr>
<tr>
<td>Constraints:</td>
<td>Non-overlap of time/room resources, Course structure requirements, Faculty time/room requirements, Class distribution ...</td>
</tr>
<tr>
<td>Objectives:</td>
<td>Minimize student conflicts, Maximize class time/room/distribution preferences</td>
</tr>
</tbody>
</table>

Problem model and constraints consider complexity of all university courses
Student Sectioning Model

Variables: Students

Domains: Assignments of students to classes

Constraints: Class limits,
            Class conflicts (overlaps in time),
            Reservations,
            Course structure,
            Course enrollment projections, ...

Objectives: Maximize satisfaction of student course/free time requests, and other preferences
Constraint Solver

Iterative Forward Search (IFS)

• Hybrid Algorithm (mixture of local search and backtracking)
• Improves upon Incomplete Feasible Assignments
• Extensible
  • Search Guiding (Meta)Heuristics
  • Dynamic Arc Consistency
  • Conflict Statistics Learning Technique
  • Dynamic Backtracking
Solution Space (LLR, Spring 07)

Number of classes: 804

Avg. number of available times: 10.6
Avg. number of available rooms: 16.3

Size of the search space \( \approx (10.6 \cdot 16.3)^{804} = 8.8 \cdot 10^{1798} \)

At 1µs per assignment, \( 3 \cdot 10^{1785} \) years to search entire space
# Experimental Results: Large Lecture Problem

- Fall 2005 data set
- Best solution within 30 minutes, 10 runs
- 1GHz Pentium III, Java 1.4.2

<table>
<thead>
<tr>
<th></th>
<th>IFS + CBS</th>
<th>Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assigned classes [%]</td>
<td>100.0 ± 0.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Time [min]</td>
<td>12.01 ± 3.77</td>
<td>~ 1 week</td>
</tr>
<tr>
<td>Satisfied student enrollments [%]</td>
<td>99.39 ± 0.01</td>
<td>98.20</td>
</tr>
<tr>
<td>Time preferences [%]</td>
<td>92.69 ± 0.34</td>
<td>89.02</td>
</tr>
<tr>
<td>Room preferences [%]</td>
<td>75.27 ± 1.42</td>
<td>83.04</td>
</tr>
<tr>
<td>Instructor back-to-back preferences [%]</td>
<td>97.29 ± 1.15</td>
<td>94.71</td>
</tr>
<tr>
<td>Departmental balancing [penalty]</td>
<td>7.60 ± 5.02</td>
<td>311</td>
</tr>
</tbody>
</table>
System Architecture

**Presentation Layer**
- User Interface
- Business Logic
- Course Structure Model
- Timetable Solutions
- All Data Persistent in Database

**Solver Implementation**
- Problem Specific Heuristics
- Timetabling Solver Data Model

**Constraint Satisfaction Layer**
- Abstract CSOP (V,D,C,f) Solver
- General Variable and Value Selection Heuristics

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

**Presentation Layer**
- Load Data
- Save Solution
- Solver Operation & Timetable Presentation
System Architecture/Interfaces

Student Information System
  Student Portal
  Event Management
  Space Inventory
  ...

Timetabling Client
  (Web Browser)

Web Server
  (J2EE Application Server)

Solver Server
  (Java Program / J2EE Service)

HTTPS

RMI/SSL

XML
Plans for Future

Timetabling:
Incorporate student course demand and free time preferences once new student systems available.
Optimize creation of individual student schedules.

Sectioning:
Real-time generation of individual student schedules after course timetable created.
Space reserved in classes at times required by late registrants (e.g., transfers and freshmen) to create full schedule.
Expected Demand in Real-Time Sectioning

Example: Course C

Course A
- Class 1
- Class 2
- Class 3

Course B
- Class 1
- Class 2
- Class 3

Course C
- Class 1
- Class 2
- Class 3

Course D
- Class 1
- Class 2
- Class 3

(Lecture w/ Grouped Lab)

Course E
- Class 1
- Class 2

Legend:
- Expected Demand
- Future Students
- Unallocated

Example: Course C (40 students)

Class 1
- Expected Demand
- Future Students
- Unallocated
System Demonstration
Instructional Planning and Scheduling Process

**Student Demand**
- Number of students
- Program of study

**Student Advising**
- Student educational goals
- Classes wanted/needed
- Eligibility to take class

**Instructional Planning**
- What classes to offer
- Spaces per class
- Who will teach each class
- Who may take the class
- Classes taken together?
- What time classes offered
- Facilities needed by class

**Class Selection**
- Class available at open time?
- Do desired classes conflict?
- Classes with multiple time offerings

**Optimization**
- Best combination of classes/times
  - Student
  - University

**Enrolling Students to Classes**

**Corrective Actions**
- More space needed?
- Enough students to teach?

**Delivery of Instruction**
- Teaching Classes
- Passing/Not Passing Student

**Management**
- Teaching loads
- Adequate instructional resources?
- Enrollment management

**Feedback**
## Timetabling/Scheduling Timeline

### Current Timeline:

| Wk -1 | Wk 1 | Wk 2 | Wk 3 | Wk 4 | Wk 5 | Wk 6 | Wk 7 | Wk 8 | Wk 9 | Sp B | Wk 10 | Wk 11 | Wk 12 | Wk 13 | Wk 14 | Wk 15 | Wk 16 |
|-------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|------|
| LLR Requests | LLR Schedule | Dept Schedules | | | | | | | | | | | | | | |
| Cur Space Req | | | | | | | | | | | | | | | |

### Potential New Timeline:

| Wk -1 | Wk 1 | Wk 2 | Wk 3 | Wk 4 | Wk 5 | Wk 6 | Wk 7 | Wk 8 | Wk 9 | Sp B | Wk 10 | Wk 11 | Wk 12 | Wk 13 | Wk 14 | Wk 15 | Wk 16 |
|-------|------|------|------|------|------|------|------|------|------|------|-------|-------|-------|-------|-------|------|
| Cur Space Req | | | | | | | | | | | | | | | | |
| List Offerings | LLR Requests | LLR Schedule | Dept/Lab Schedules |
| Student Preliminary Schedule Requests | Continued Requests | | |
| | | | Real-Time Scheduling |
The End