Purdue University Academic Scheduling

8 January 2007

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

Why Develop Scheduling Systems?

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

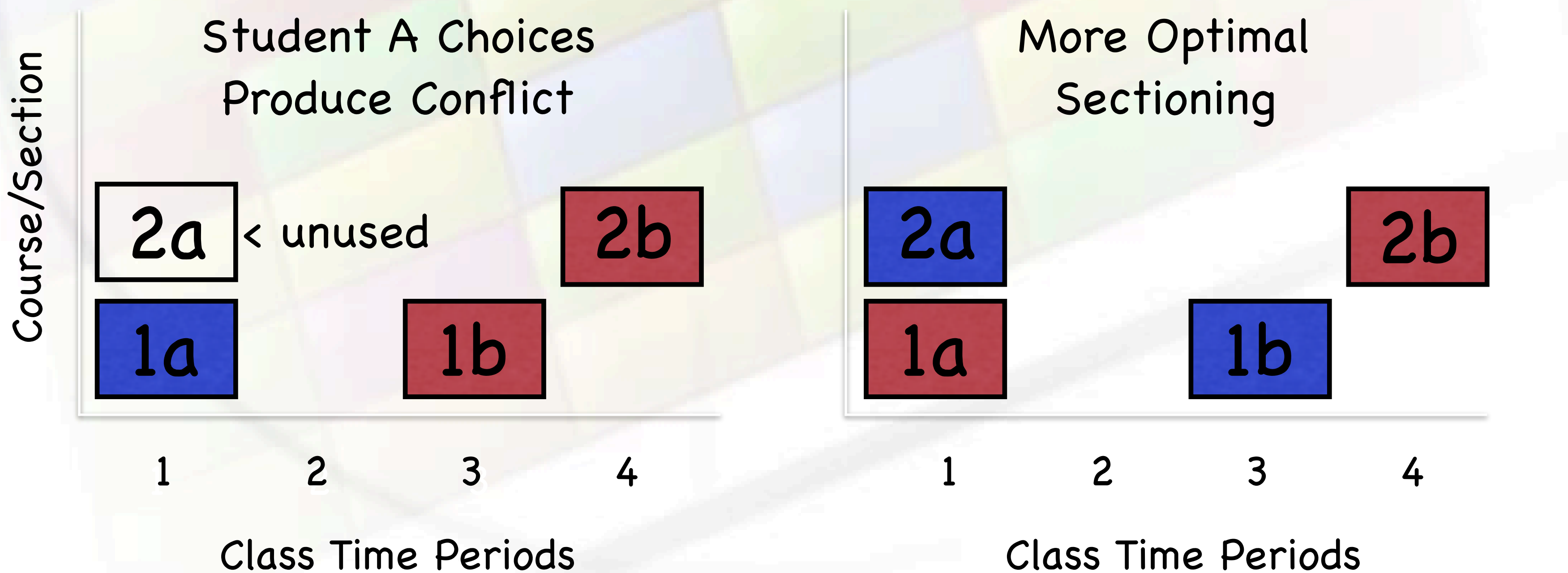
3. Need for tool to modify existing timetable (class limit changes, ADA accommodations, etc.)

4. No additional resources to support enrollments $\Rightarrow$ 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:



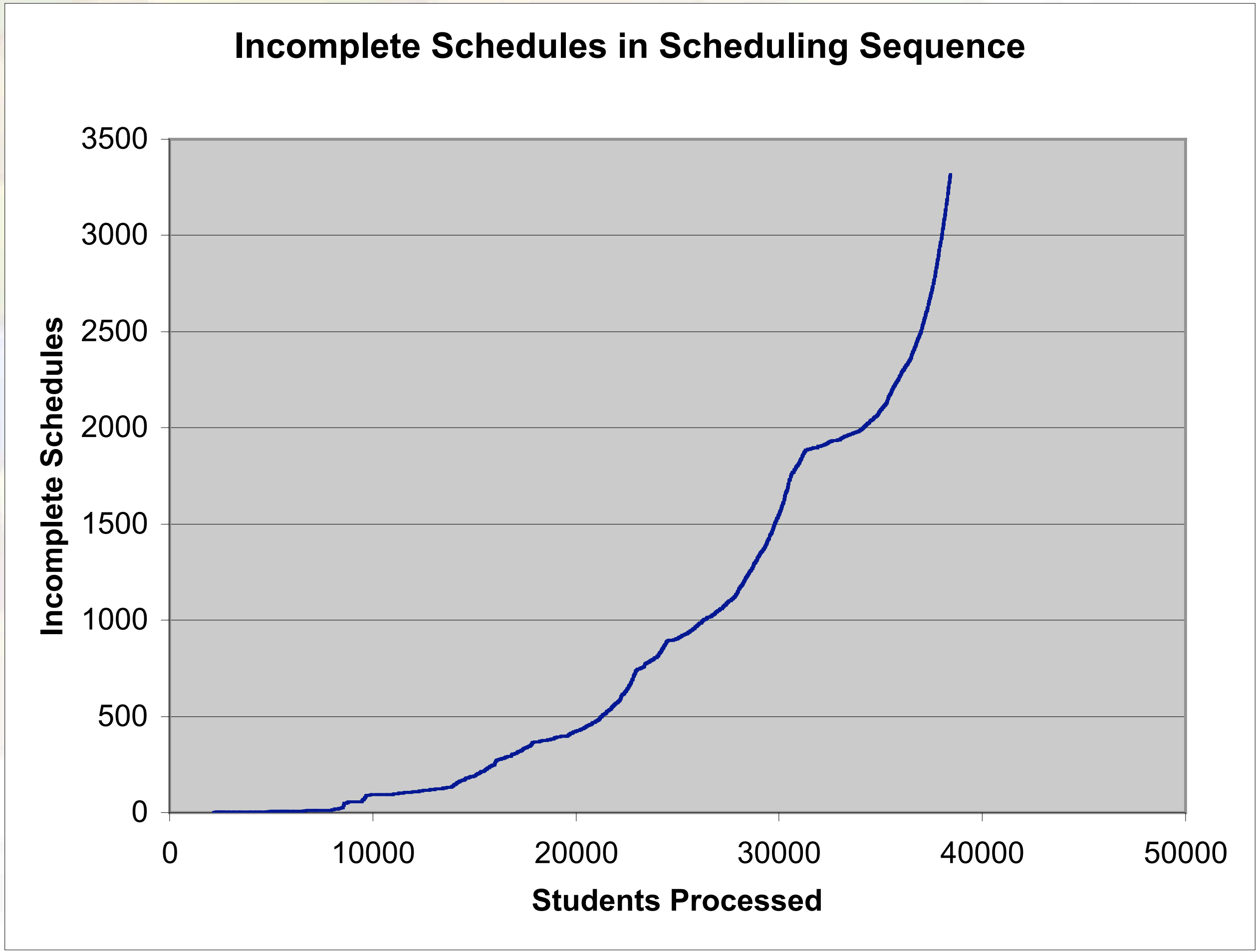
Need for Student Sectioning: Experiments

Seniority Order Registration with Time Selection

# of Classes not Scheduled	Students w/ Unscheduled Classes
1	2693
2	513
3	94
4	16
5	0
	3316

Fall '04

Requests	38429



Project Goals

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
  - $\eta \subseteq \{ v/a \mid v \in V \ \&\ a \in Dv \}$
  - 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

Constraint Satisfaction Optimization Problem

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
    - $f(\eta)$ = (weighted) sum of soft constraints that are not satisfied

Solution

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

Course Timetabling Model

Variables: Classes

Domains: Values of time and room assignments

Constraints: Non-overlap of time/room resources,
Course structure requirements,
Faculty time/room requirements
Class distribution ...

Objectives: Minimize student conflicts,
Maximize class time/room/distribution preferences

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, ...

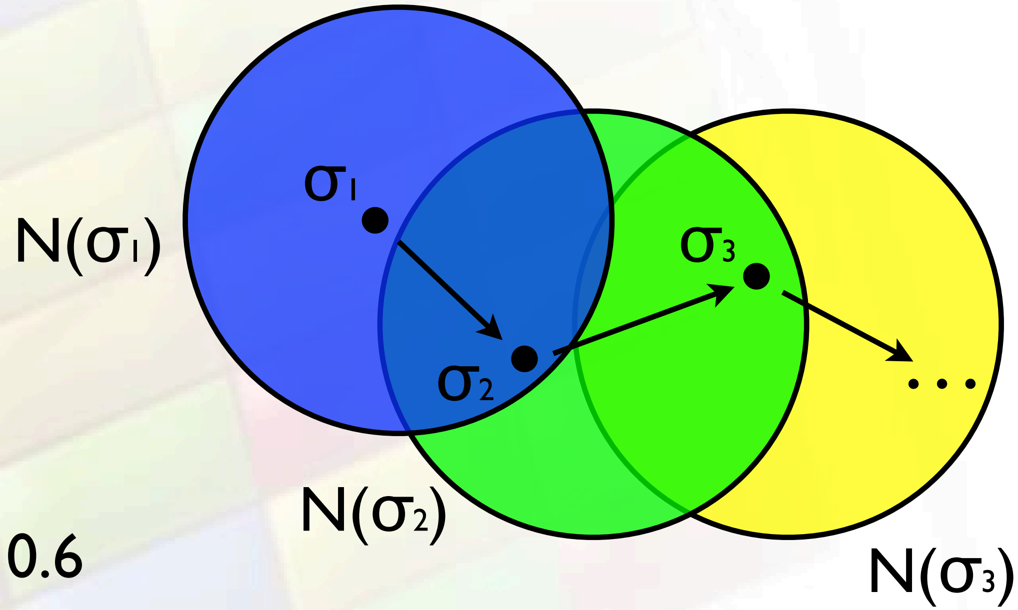
Objectives: Maximize satisfaction of student course/free time requests, and other preferences

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 $\approx (10.6 \cdot 16.3)^{804} = 8.8 \cdot 10^{1798}$

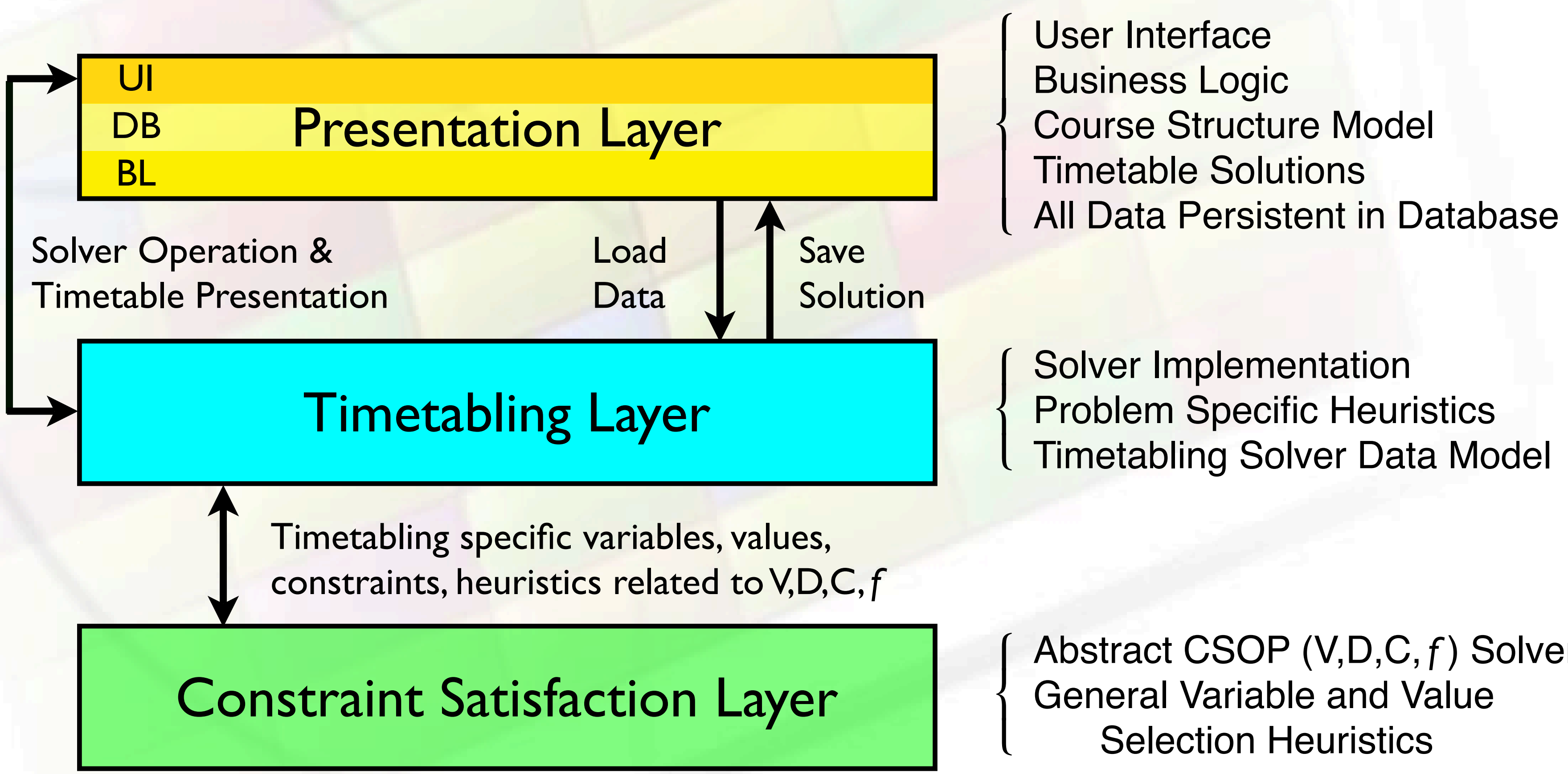
At 1μs per assignment, $3 \cdot 10^{1785}$ years to search entire space

Experimental Results: Large Lecture Problem

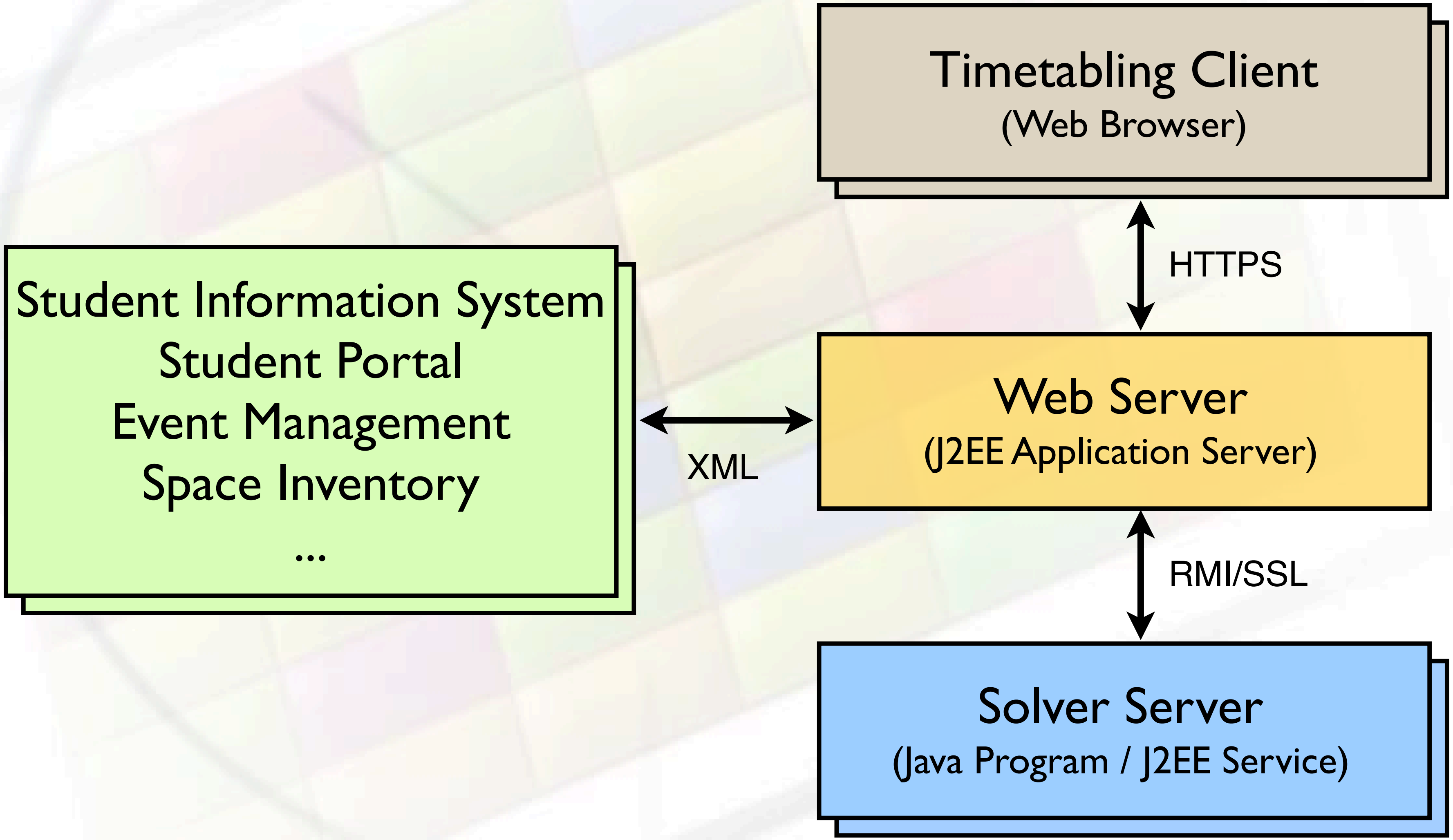
- Fall 2005 data set
- Best solution within 30 minutes, 10 runs
- 1GHz Pentium III, Java 1.4.2

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

System Architecture



System Architecture/Interfaces



Plans for Future

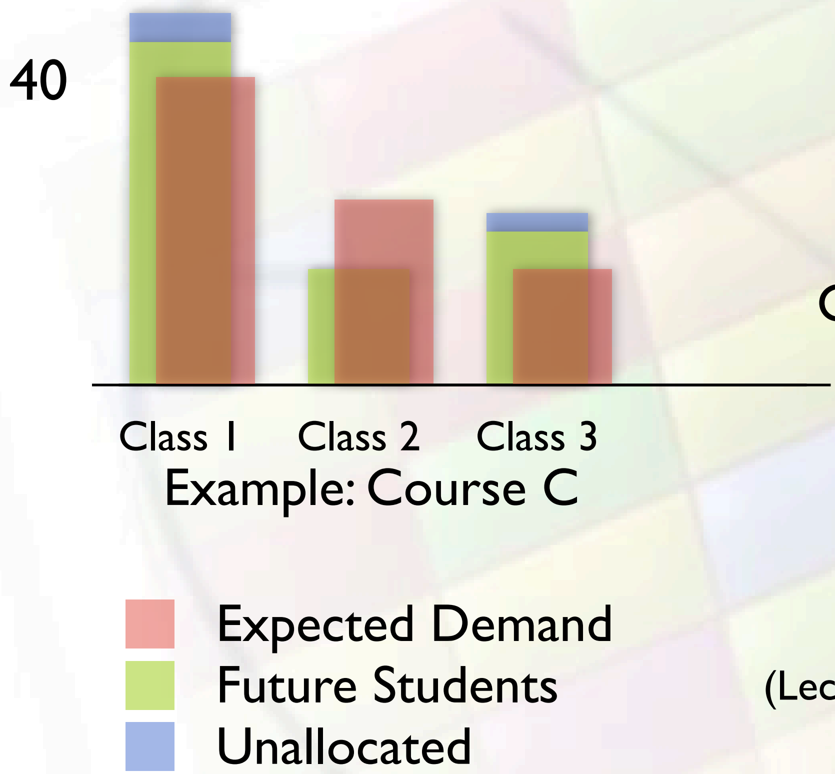
Timetabling:

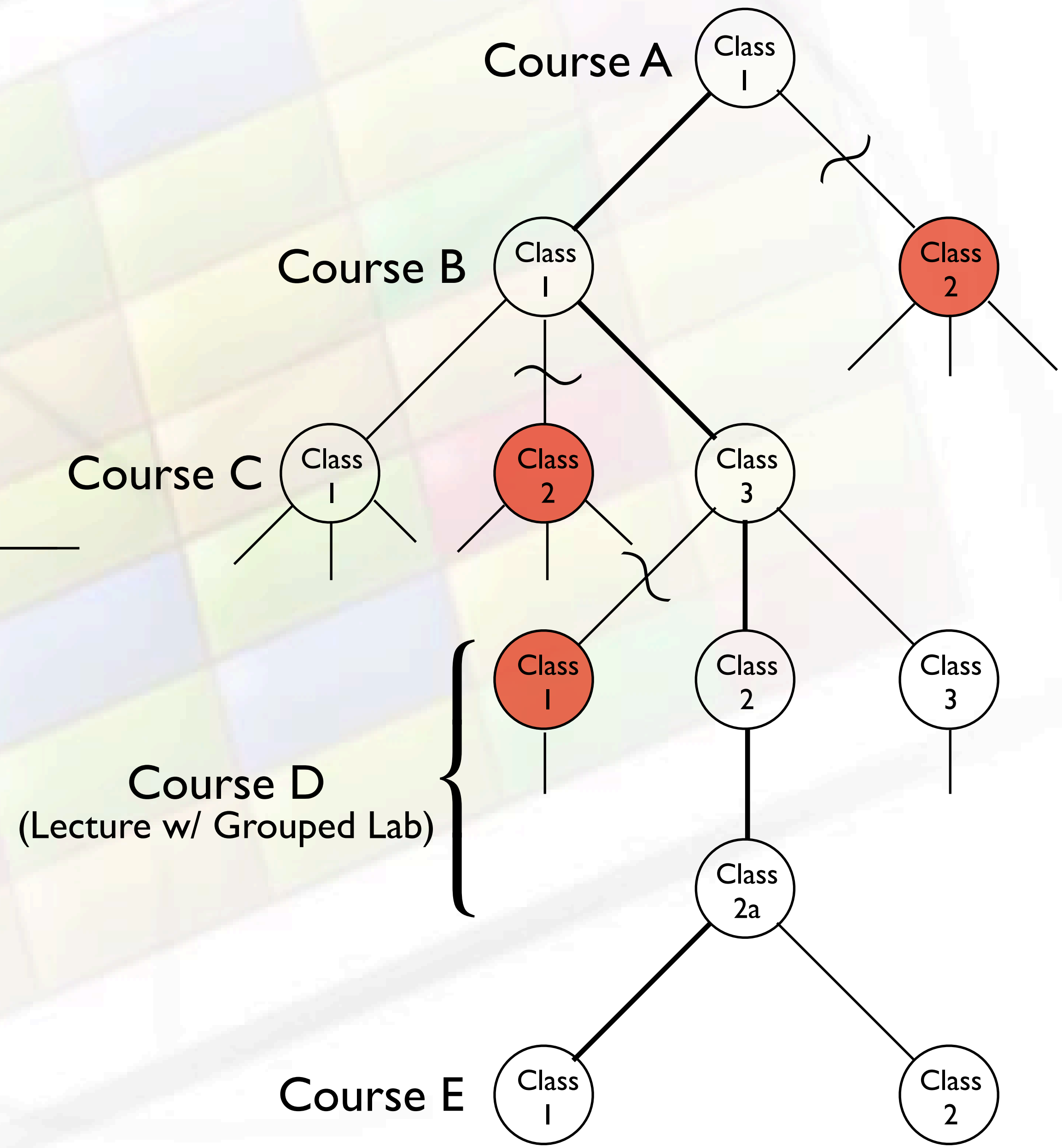
- Incorporate student course demand and free time preferences once new student systems available.
- Optimize creation of individual student schedules.

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





System Demonstration

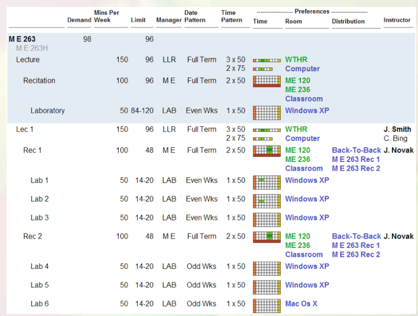
Instructional Planning and Scheduling Process

Student Demand

Number of students
Program of study

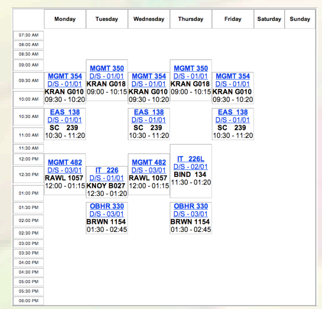
Student Advising

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					Student Registration								
	Cur Space Req													Schedule Runs			

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						LLR Requests		LLR Schedule		Dept/Lab Schedules						
	List Offerings		Student Preliminary Schedule Requests							Continued Requests				Real-Time Scheduling			

The End