UNI time.org

PURDUE UNIVERSITY
Space Management & Academic Scheduling

Real-time Student Sectioning

Keith Murray
Tomáš Müller

MISTA 2007

Problem Summary

Student Sectioning:

Assignment of students to specific sections of requested courses while minimizing time conflicts

- Often be viewed as subproblem of course timetabling
- Student section assignments optimized during timetable construction or as follow-on process

Student Expectations:

Support student self-service class scheduling in real time

- No patience for batch process to optimize section assignments
- Students want some options for choice of class time

Real-time Sectioning

Minimizing conflicts during a real-time scheduling process

- In practice, student choices may produce conflicts for later students

- Schedulers may adapt by holding sections or spaces in sections for later use by students needing specific times

Optimizing Real-time Sectioning

Can the process of reserving space in sections for later students be automated and assignment of students to these spaces be optimized during real-time sectioning?

Given:

- Student course demand combinations are known (at least approximately) and available for construction of the timetable

Hypothesis:

- Information about course demand patterns can be used to improve a real-time decision process for student section assignment

Real-time Sectioning

Proposed Process:

Identify expected demand for each class section using

- historical course combinations required by students
- placement in the existing timetable

Limit sections available to students based on the expected demand for each section versus the space available

Automated Timetabling and Batch Sectioning

Current (or historical) student course demand assists in determining best time placements of class sections in the timetable

Batch sectioning process at the end of timetabling determines amount of space available in each class section for real-time sectioning

- Creates best fit of existing and expected students to classes to minimize student conflicts
- Existing (pre-registered) students receive individual class schedules from batch sectioning in timetabling
- Spaces assigned based on last-like semester student data are reserved for expected later registrants

Batch Sectioning Process

Solver uses Iterative Forward Search algorithm

- Local search based, works with incomplete feasible solutions, extensible

Variable: Student request for course or free time (including alternatives)

Value: Assignment of student into the course (selection of sections)

Constraints: Course structure, limits, reservations, section times (overlaps), and rooms (distances between back-to-back sections)

Real-time Sectioning Process

One student is processed at a time

- Returning a full (or most complete) schedule
  - Considers priorities, alternative courses, free-time requests
  - Includes choices of available classes and possibility to wait-list filled classes

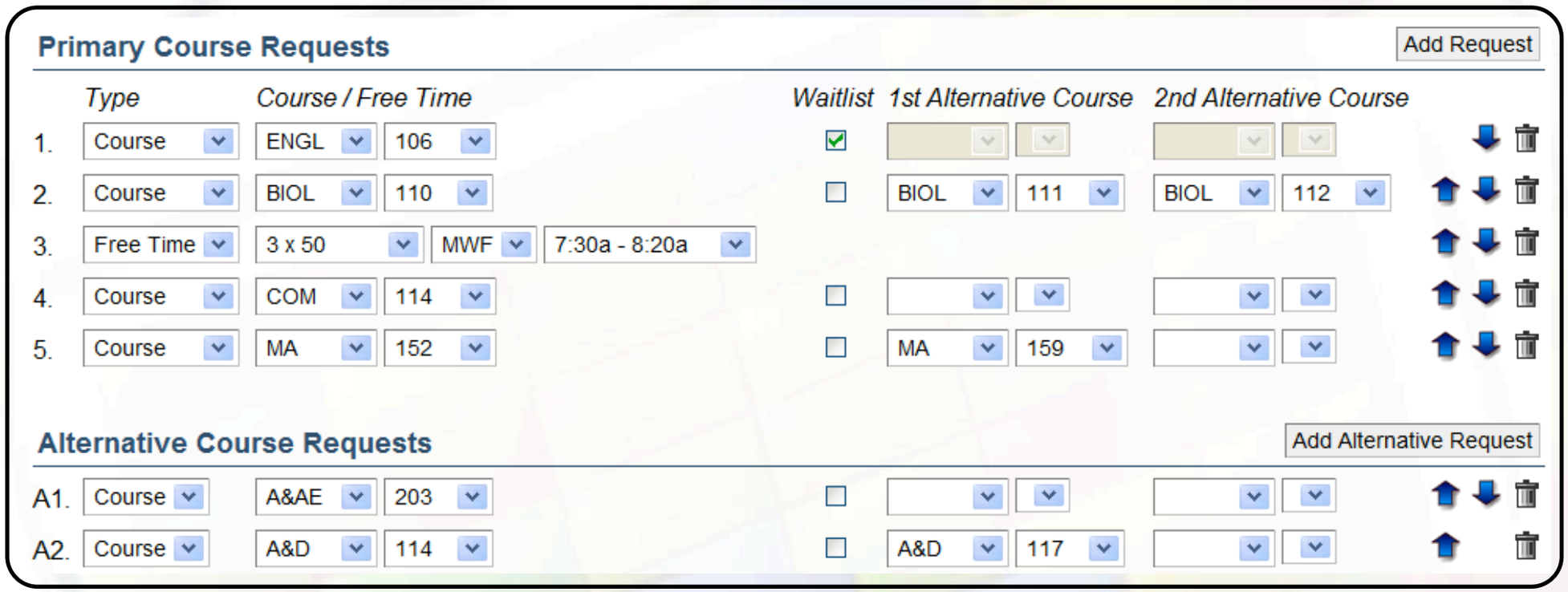
Uses same model as batch sectioning, different algorithm

- Branch & bound returning most complete schedule
- Optimizing given preferences (free times, alternatives)
- Avoiding sections reserved for expected students

Incoming students (freshmen)

- Reservations for expected students are used and updated as students are enrolled into classes

Student Course Request Model



- Students submit course and free time requests, including priorities and any alternates
  - Pre-timetabling requests are used by timetabling solver to minimize student conflicts in class time placement and batch sectioning
  - Post-timetabling requests and change requests use real-time sectioning process

Real-time Sectioning: Algorithm

Branch & bound with the following rules (goals):

- Return the most complete schedule
- If there is a choice of enrollment between two courses, the one with the higher priority must be selected
  - An alternative course can only be used when the student cannot be enrolled in the primary course (e.g., because of a conflict with a higher priority course)
- Free-time requests are treated as ordinary course requests
  - Courses of lower priority cannot overlap requested free times, courses of higher priority may (but schedule is less complete)
  - Except when the solver would be forced to pick a section that is fully reserved for incoming students (even if an alternative course is available)
- Distance conflicts are allowed, but minimized
- Courses are ordered by student's priority (variable ordering)
- Enrollments are ordered by currently available space in sections (value ordering)

Real-time Sectioning: Section Weight

Computation of spaces to reserve (two counters on each section)

- Held space: Number of spaces assigned to expected future students in a section
- Expected need: Projected number of spaces required in a section by expected students
  - For each expected student (with enrollments to other courses fixed):
    - Number of possible enrollments into the course containing this section,
    - Divided by total possible enrollments to sections in this course.
  - E.g., if a student can attend either Lec 1 or Lec 2, ½ is added to both lectures.

Section Weight

- Existing students: Based on difference between currently available and held space
  - Directs assignments away from sections expected to be taken by later students
- New students: Based on difference between currently available space and expected need
  - Directs section assignment away from sections most likely to be required by other students expected to enroll in the future

Update (after each new student)

- Held space: -1 for section enrolled by the student
- Expected need: decreased by the expectation that a student could attend the section

Practical Results

Fall 2007 course timetabling data from Purdue University

- 9,000 classes, 570 rooms, 39,000 students with 190,000 course requests

Student Data

- Last-like semester (fall 2006) student course requests
  - 185,494 course requests from 38,740 students
- Real (fall 2007, as of July 11) student course requests
  - 187,847 course requests from 36,117 students
- No alternatives, free time requests, or course reservations
- Invalid course requests were eliminated (~2,000 requests)

Test Cases

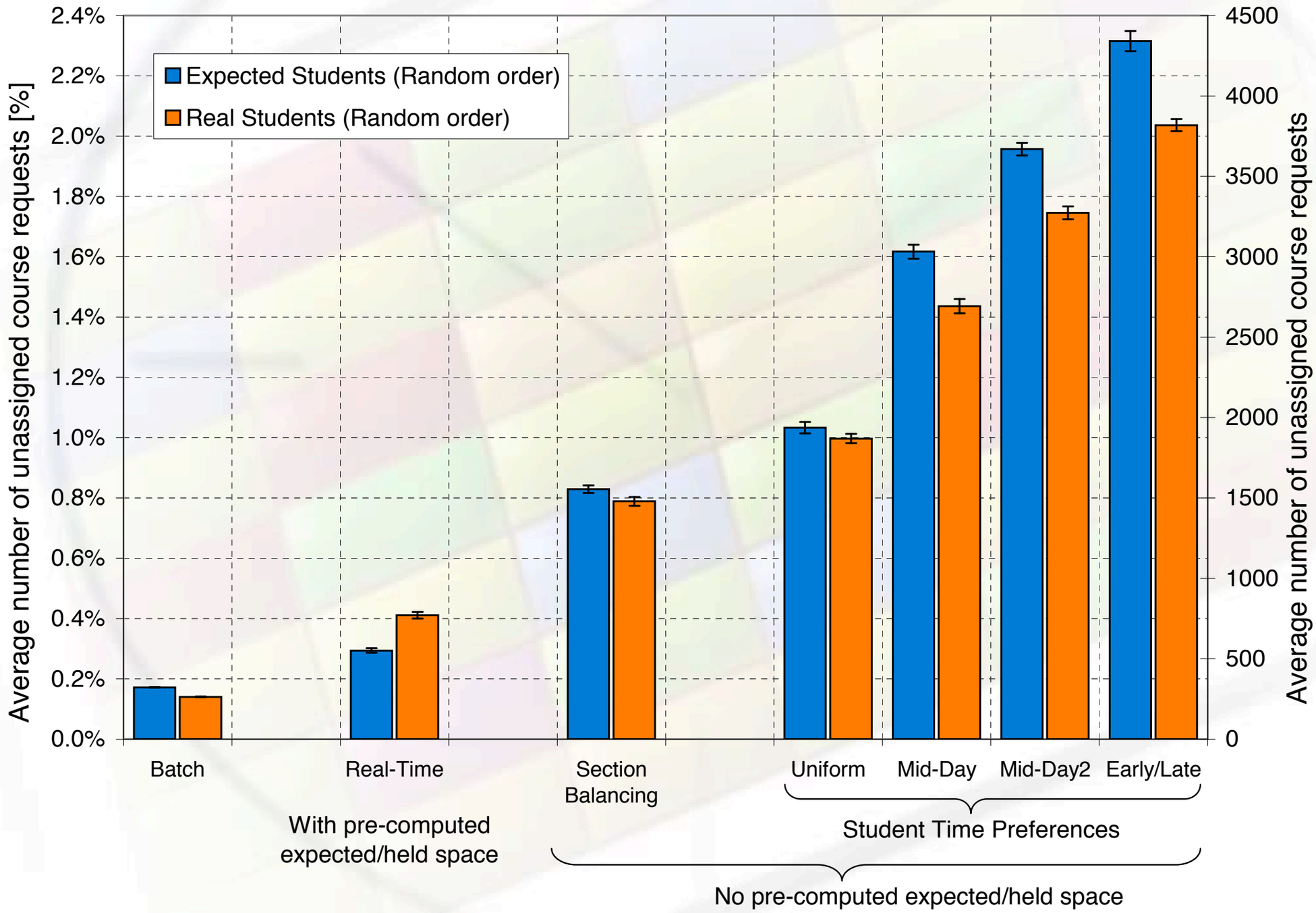
- Batch sectioning of last-like semester student course requests
  - Results used to calculate held spaces and expected need for real-time sectioning
- Batch Sectioning of real student course requests
- Real-time sectioning of expected and real student course requests
  - Students are sectioned one-by-one in a random (or given) order
- Sectioning by student time preferences (uniform, mid-day, early/late)

Practical Results: Batch vs. Real-time

- Fall 2007 course timetable, fall 2006/2007 student course requests
- Average and RMS values from 10 independent runs
- Number and percentage of unassigned course requests
- For online sectioning runs, expected need and held space computed from best last-like semester batch run (316 unassigned requests)

	Expected Students	Real Students
Batch Sectioning (Average ± RMS of 10 runs)	0.171% ± 0.001% (317.92 ± 1.58)	0.140% ± 0.001% (263.17 ± 1.97)
Real-time Sectioning (Random order)	0.294% ± 0.007% (545.20 ± 13.54)	0.411% ± 0.011% (772.45 ± 20.34)
Real-time Sectioning (Students with more choices first)	0.401% ± 0.007% (744.00 ± 12.46)	0.556% ± 0.008% (1043.70 ± 15.00)
Real-time Sectioning (Students with less choices first)	0.249% ± 0.002% (461.57 ± 3.59)	0.318% ± 0.002% (597.00 ± 4.27)
Real-time Sect./Section Balancing (Random order, no expected/held)	0.829% ± 0.013% (1538.07 ± 23.78)	0.789% ± 0.015% (1481.90 ± 27.39)

Practical Results: Unassigned Course Requests



Conclusions

It is possible to significantly improve the quality of class schedules that can be produced for students by incorporating known course demand data during a real-time sectioning process

- Can not meet student course requests as well as batch sectioning, but results much better than when knowledge of required course combinations not considered
- Non-uniform student time preferences, or preferences that are inconsistent with the timetable decrease ability to meet student course needs during the sectioning process
- Ordering of students can affect results: processing students with greatest number of section choices first results in more unassigned requests

Proposed Application: Student Class Schedules

Combine Batch and Real-time Reservations

- On courses or sections
- Academic area, major/minor, learning communities, individual

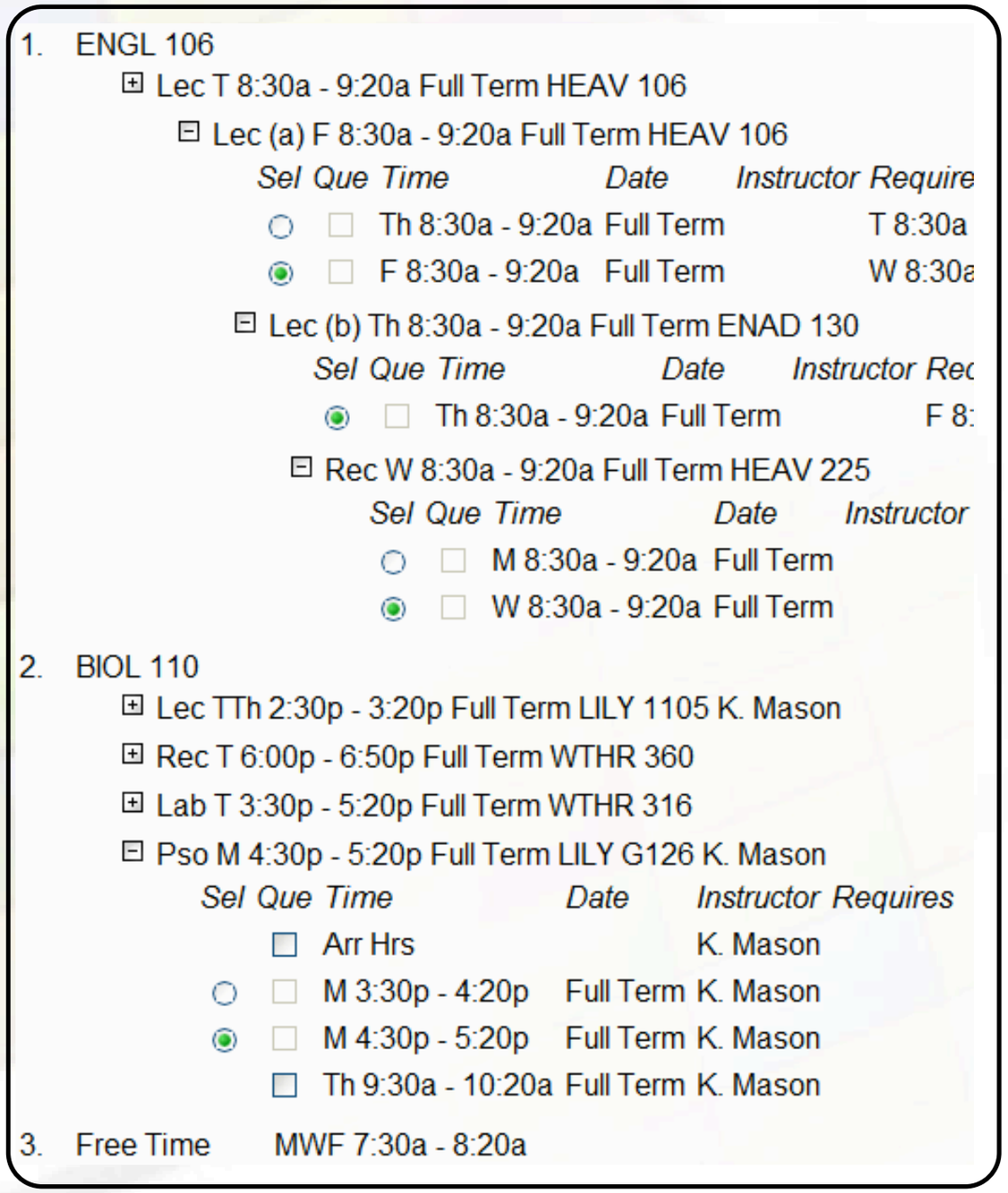
Wait-listing on courses

Re-sectioning

- Wait-list processing, timetable changes, mass cancellations, etc.

Student choices

- (Limited) ability to choose time and instructor
- Choice between available sections
- Wait-list for unavailable sections
- Interactive schedule changes



Additional Information: UNI|Time Web Site

Available for download

- Course timetabling and student sectioning application
  - Open source (GNU GPL)
- Constraint solver library
  - Including timetabling and sectioning extensions
  - Open source (GNU LGPL)

Online documentation

Ongoing research

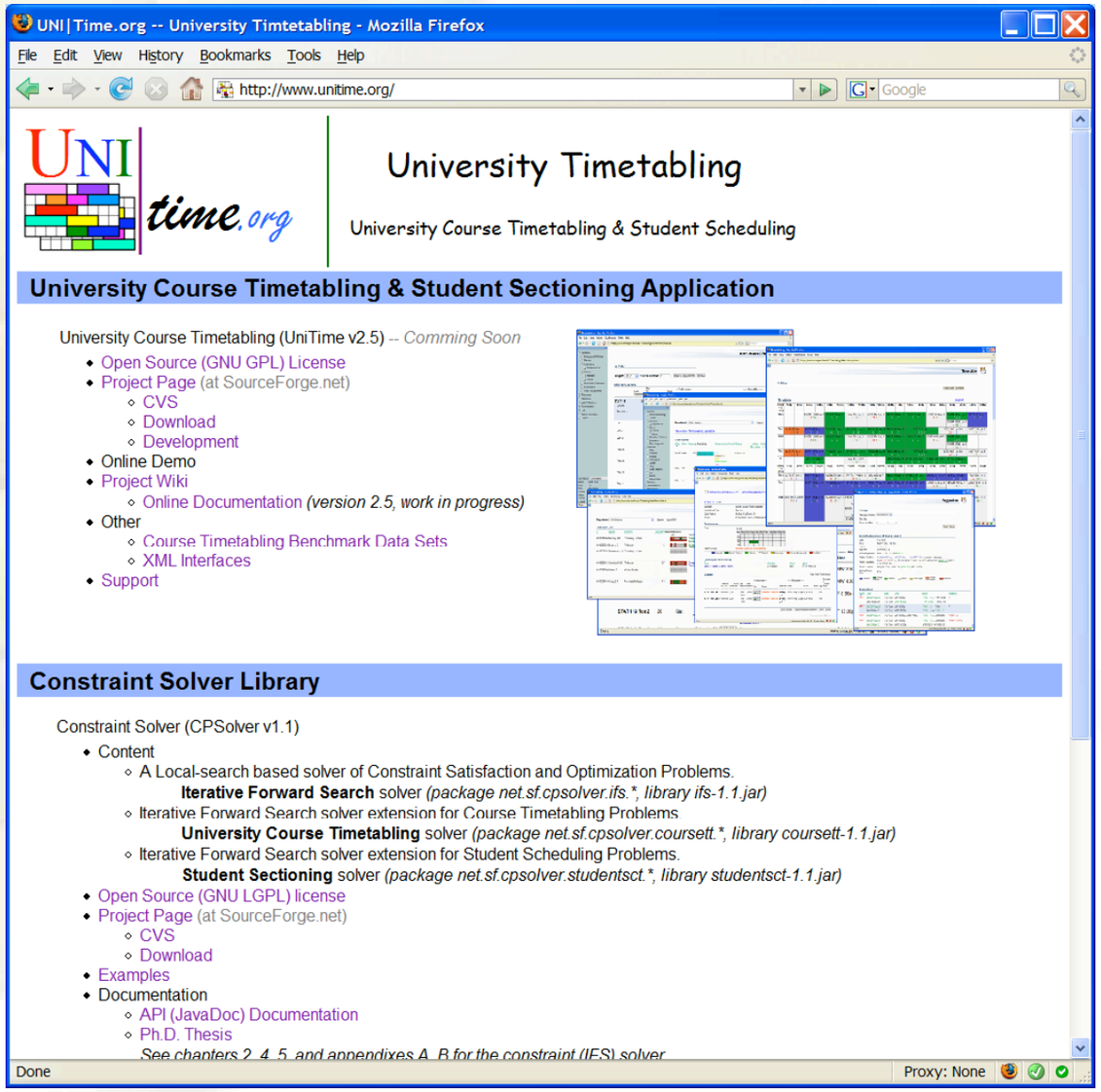
- Publications and presentations

Benchmark data sets

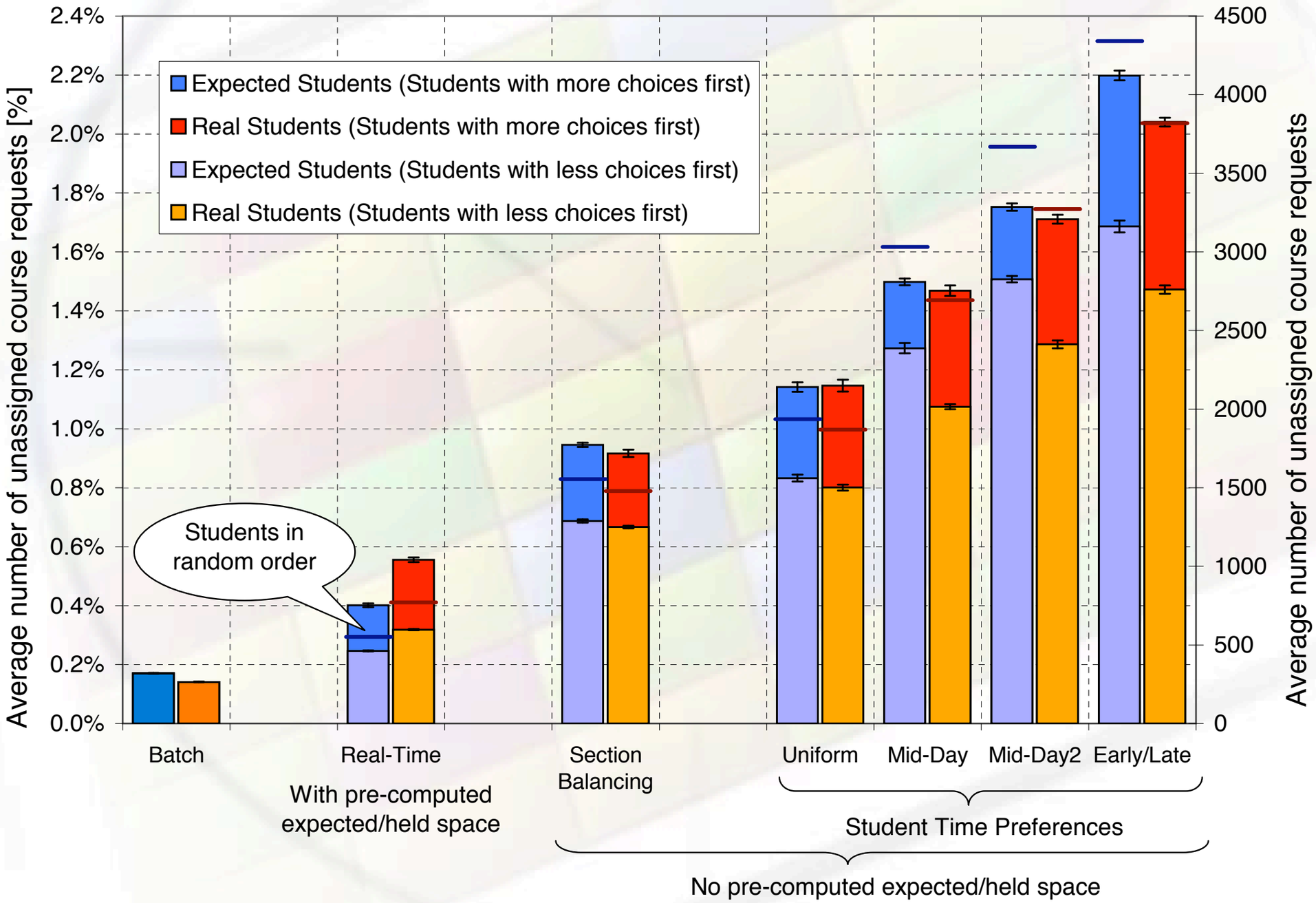
- Real-life data for course timetabling and student sectioning problems

Contact: research@unitime.org

URL: http://www.unitime.org



Practical Results: Ordering of Students



Practical Results: Student Time Preferences

For each student, all possible times are ordered using Uniform, Mid-Day, Mid-Day2, or Early/Late preferences

- For a student, the most complete schedule is selected
- Minimizing distance conflicts and optimizing time preferences

Mid-Day Example: Time is divided into one hour blocks

- Each block has weight between 1 and 10
- All other times (before 7:30 am, after 5:30 pm, weekend) have weight 1

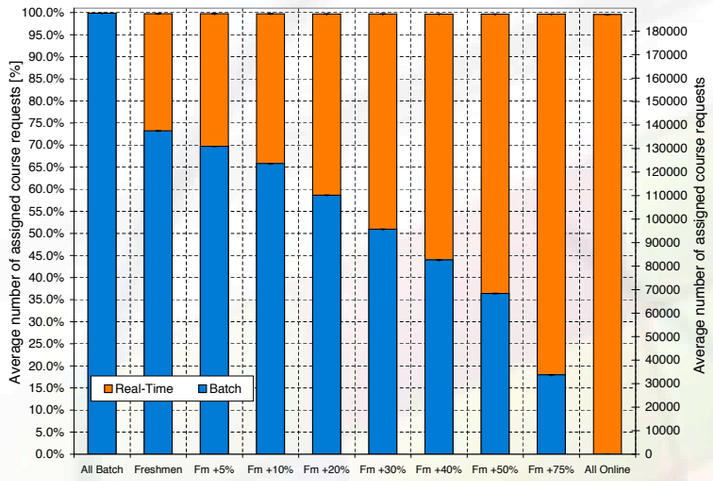
	7:30a	8:30a	9:30a	10:30a	11:30a	12:30p	1:30p	2:30p	3:30p	4:30p
Mon	1	4	7	10	10	5	8	8	6	3
Tue	2	4	7	10	10	5	8	8	6	3
Wed	2	4	7	10	10	5	8	8	6	3
Thu	2	4	7	10	10	5	8	8	6	3
Fri	2	4	7	10	10	5	4	3	2	1

Other: $\text{Weight (time)}_{\text{Uniform}} = 1$, $\text{Weight (time)}_{\text{Early/Late}} = 11 - \text{Weight (time)}_{\text{Mid-Day}}$

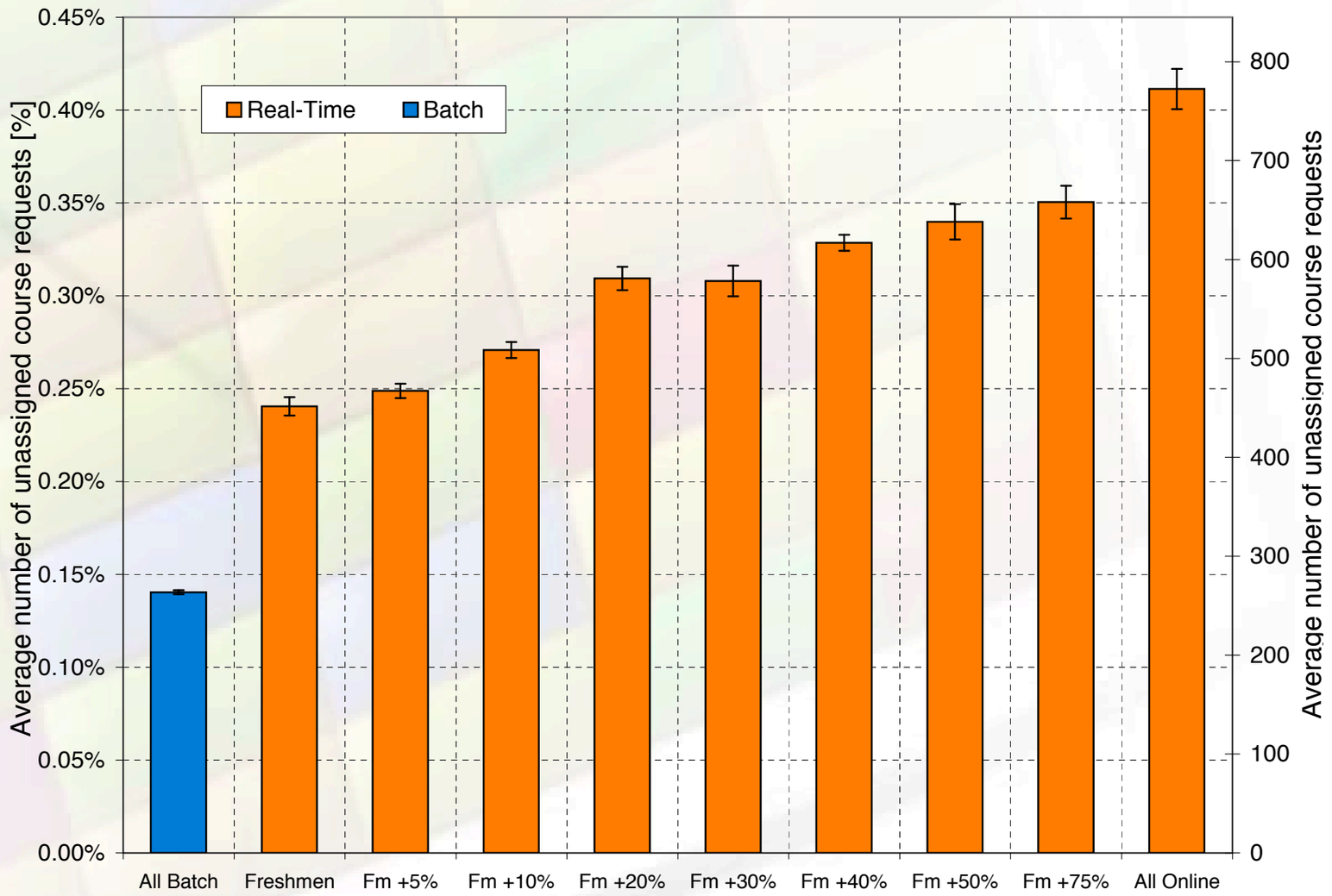
$\text{Weight (time)}_{\text{Mid-Day2}} = [\text{Weight (time)}_{\text{Mid-Day}}]^2$

For each student, an ordered set of time blocks is created by repeated drawing of time blocks based on their weights (roulette-wheel)

Combining Batch and Real-Time Sectioning



% of Course Requests Assigned using Batch vs. Real-time Sectioning



Unassigned Course Requests