IE511: Integer Programming
Spring, 2014
Instructor: Prof. R.S. Sreenivas
155 Coordinated Science Laboratory (Primary Office)
110 Transportation Building (Secondary Office)
e-mail: rsree@illinois.edu
Tue-Thu, 10:00-11:50AM, 170 Everitt

Course Description: This course is about the optimization of linear systems involving discrete variables. After introducing Integer Linear Programs (ILPs) via examples, we will review the key results in Complexity Theory. We will identify conditions that yield tractable families of ILPs, and we will look at algorithms that solve the general ILP instance. Some programming experience is desirable, but not required.

Primary Text: Lessons with Code Samples written by me.

1 Tentative Syllabus

“You must fill your heads with wisdom before you can break boards with it.”

— Karate instructor on “The Simpsons”.

Lectures

1. Introduction.

   (a) Review of – Linear Algebra; Systems of Inequalities; Convex Hulls; Polyhedra; LPs.

   (b) Integer Linear Program Examples ; Existence of a Unique Solution, Examples.

2. Polyhedral Theory.

   (a) Polyhedra and Integer Programs: Valid Inequalities and Faces of a Polyhedra, Dimension, Extreme Points, Facets, Minkowski’s Theorem, (Most) IPs are LPs.

   (b) Integrality of Polyhedra: Total Unimodularity, Matchings and Integral Polyhedra, Total Dual Integrality, Submodularity and Matroids

3. Algorithms (Tentative)

   (a) Complexity Theory: P, NP, PSPACE and EXPSPACE classes, complexity of IP problems, Optimization and Separation, HORNSAT Polytopes, Quadratic-SAT (QSAT) Polytope.

   (b) Relaxation and Bounds: Relaxation and Optimality, Combinatorial Relaxations, Lagrangian Relaxation, Duality.
(c) **Dynamic Programming**: Shortest paths, Knapsack problems, problems on Trees.

(d) **Branch and Bound**: Divide and Conquer, Pruning Enumeration Trees, LP-based Branch and Bound.

(e) **Cutting Planes**: Cutting-Plane Proofs, Chvatal Rank, Gomory’s Cutting-Plane Algorithm.

(f) **Approximate Algorithms & Heuristics**: Christofides Algorithm for Symmetric TSP with Triangle Inequalities; The Swap and Reversal Heuristics for TSP Instances; Approximation Algorithms for Scheduling; Metric Uncapacitated Facility Location Problem by LP-rounding; Approximate solutions to Knapsack Problems; The MAX-SAT Problem and its Approximations.

**Programming Assignments** I would like everyone to become proficient in C++ programming by the end of this course. Do not worry if you have not programmed before. I will provide the necessary starting material and sample code that you can develop further. I will use a public domain MILP solver called lp.solve, and you will work with its API. I plan to have five programming assignments that involve writing solvers for different ILP-instances to be announced later.

## 2 Grade Composition

- Homework Assignments (30%)
- Programming Assignments (30%)
- Mid-Term Exam (20%)
- Final Exam (20%)

## 3 General Instructions

I plan to have \( \approx 5 \) homework assignments for this course. Since the pace of the course will be dictated by the class needs/skills, the exact number of these assignments might vary. The contribution to the grade will be 30% independent of the number.

There will be \( \approx 6 \) programming assignments for this course. You must turn-in an electronic version of your code on or before the date they are due. Please do not ask for extensions in the 11th hour. You will get full-credit if your submitted code works when compiled and run on a data-set of my choice. We will revert back to you if there is an error/problem with your submission. You then have three days to turn-in a corrected version, at the loss of 20 points. This process is repeated at most two times (i.e. inclusive of your first attempt,)
you have three chances at getting the programming assignment right). The programming assignments will contribute 30% towards your final grade.

The mid-term and final examinations will be in-class written exams.

I intend to use the ±-grading system. If everything goes as planned, my lecture notes for the course can be found on the University of Illinois’ Compass website. I suggest you print the appropriate lesson before class and follow-along. This will free you from the tedium of copying material off the board during class, you can use that time to follow the material presented in class instead. It is your responsibility to check the above URL regularly for updates/due-date-announcements as the course progresses.

I will be happy to speak to you if you are having trouble with the programming-aspects of the course material. I prefer to work on an “interrupt-driven” mode – as opposed to having a fixed-meeting time each week. If you need to speak to me for any reason, just send me an e-mail, and I will let you know the times when I can meet with you. I will let you know where, among my two offices, we will meet.

Good luck and I am looking forward to seeing you do well in this course!