Title: Recent Directions in Algorithms (CSE 203)

Instructor: Russell Impagliazzo

Time: MW 5-6:20

Textbooks: Motwani and Raghavan, Randomized Algorithms Vazirani, Approximation Algorithms, CLRS (back-up) Note: abbreviated MR, V, CLRS.

Scope: The course will cover algorithmic concepts and techniques from the last twenty years, with the goal of reaching the research frontier in algorithms. It will try to be self-contained, beyond a basic understanding of algorithms (i.e., CSE 202), and to be of interest to most researchers in computer science, not just theorists. There are three main related themes to the course: 1. Approximation algorithms. 2. Randomized algorithms. 3. Linear and convex programming. Research frontier topics are asterisked in the following:

1. Background: Probability theory. Read 441-446, of Motwani-Raghavan, starting with "discrete random variable". OR V. App. B
2. Background: Classification of probabilistic algorithms: MR Section 1.2 (pp. 9-10) and 1.5 (pp. 18-23). OR V. A.4
3. Background: Algebra and number theory, MR pp. 395-402.
4. Basic techniques for designing and analyzing approximation algorithms: approx. 5 lectures
   (a) Approximation-preserving reductions, e.g. Metric TSP (V., Chapter 3.2)
   (b) Reduction in precision, e.g., Knapsack (V. Ch 8).
   (c) The "Achieves a linear combination of bounds" method, e.g., minimum makespan (V, Ch. 10), set cover (V,Ch.2), weighted vertex cover (V, Ch. 2)
   (d) The "Each step only makes it a little worse" method. (e.g., interval scheduling)
5. Basic techniques for designing and analyzing randomized algorithms: approx 4 lectures
   (a) The "abundance of witnesses" method. (Chapter 14, MR) Prime testing. Finding primes. Modular square roots. Algebraic circuit zero-testing.
   (b) The "achieves the average" method. Max bisection, Game-tree evaluation (MR pp. 28-31).
   (c) The "sample reveals the whole" method. Mincut, (MR, 1.1., and Karger paper.)
6. * Approximation algorithm for Euclidean TSP (V, 11) (2 lectures)
7. Linear and convex programming (6 lectures)

(a) Linear programming basics (CLRS Ch. 29, MR, Ch. 9.10, V Ch 12)
(b) The dual of a program
(c) The ellipsoid algorithm (not in texts, see Papadimitriou and Stieglitz, Combinatorial Optimization)
(d) Approximation algorithms based on linear or geometric relaxation: rounding, primal-dual, randomized rounding, and * semi-definite programming. (V, Ch. 13-16, 26)

8. On-line and streaming algorithms (MR, Ch. 13)
9. Markov Chain Monte Carlo methods (MR Ch6, 11; V, Ch 28)

Exact topics may change due to student interest.

Prerequisites: CSE 202 or the equivalent will be assumed as a prerequisite.

Assignments: There will be two homework assignments (15 % each) e, a take-home final (50 %), and a project (20 %).

The homework and project can be done in a group of up to 3. The final must be individual effort.

You must cite any person, book, paper, document or web-site that you consult in doing assignments, other than myself and the textbooks above. Work without relevant citations is plagiarism. Work with citations is not, but it may not receive full credit if you did not solve problems yourself.

The homework assignments will be theoretical in nature, designing, analyzing and proving algorithms correct.

The project is to read and, to some extent, extend, a recent (since 1990) research paper in algorithm design or applications of algorithms. Extending may be to provide additional analysis, to give a stronger analysis for a special case of input, or to implement an algorithm and obtain experimental performance results. This project may be related to your own research or to projects for other classes, e.g, a computational bio class. However, you must cite the related work for other classes, and rewrite the project to emphasize the use of algorithms. Suggested length is 5-8 pages.