CS 590-AAA Approximation Algorithms in Action
Spring 2021, Tues Thurs 4:30 – 5:45 PM On line
Alex Pothen, apothen@purdue.edu
www.cs.purdue.edu/homes/apothen/Courses/CS590-AAA.pdf

Syllabus

Introduction. Massive graphs are ubiquitous in many fields of computing, science, engineering and industry. The human brain contains trillions of neurons and synapses. Massive sources of data such as internet traffic logs contain trillions of events, e.g., the CAIDA internet "telescope" at UC San Diego, and these can be represented, visualized and modeled by graphs. Algorithms for graphs with billions or more edges are needed to solve many computational problems, but unfortunately many problems on graphs do not have polynomial time algorithms. Even when such algorithms exist, they could be impractically slow, requiring hundreds of hours or more, and do not scale to large graphs.

One remedy for the above problem is to seek algorithms that solve the problem approximately. When we can obtain worst-case approximation ratios for an algorithm over all graphs for a problem, we have an approximation algorithm. Such algorithms often have modest time requirements, with time complexity nearly linear in the size of the graph. Approximation algorithms could also be designed to have greater concurrency, so that parallel algorithms with linear work and logarithmic depth can be designed.

In this course we will study the design of approximation algorithms for several optimization problems on graphs. We will consider several variants of the matching problem in graphs, including maximization versions of cardinality matching, edge-weighted matching, edge-weighted b-matching, vertex-weighted matching, and stable matching. We will also study the related edge cover and b-edge cover problems. Vertex cover, maximum independent set, and vertex coloring are other problems that we will consider. Our focus would be on practical algorithms that could be implemented to obtain good performance on modern multi-core processors available on desktops and laptops.

For each problem, we will explore the underlying combinatorial framework that enables the design of approximation algorithms. These include matroids, which are combinatorial structures that generalize graphs and matrices, and relaxations of matroids, such as k-extendible systems. The primal dual linear programming framework will play a central role, as well as randomization, and more recent algorithms for optimizing submodular functions.

I gave a talk on some of these topics in a seminar at MIT, and you can see the video here:
www.dropbox.com/s/ocf9rkwf0bbngd0/MIT-CSAIL-FastCodeSeminar-Pothen-10-26-2020.mp4?dl=0

List of topics. We will plan to discuss the following topics organized in terms of broad unifying themes:

- **Introduction and motivation**. The central role of matching problems and algorithms in computer science and combinatorial optimization. Matching and edge cover problems that we will consider. Brief background material on: graphs,matroids, linear programs, parallel algorithms (shared-memory and distributed-memory models), randomized algorithms. The approximation landscape: PTAS, FPTAS, inapproximability, etc. Algorithm Engineering, high performance graph algorithms.
- **Combinatorial structures**
  - Matroids, maximum cardinality and vertex-weighted matching.
  - Matroid intersection, maximum matching in bipartite graphs
  - Matroid relaxations and k-extendible systems, edge-weighted b-matching
  - Submodular constraints and b-edge cover.
- **Techniques for exact and approximation algorithm design**

- Greedy algorithms and variants: exact algorithms for vertex-weighted matching, 1/2-approximation for edge-weighted matching, 1/3-approximation for submodular matching, 3/2-approximation for b-edge cover.
- Augmentation based algorithms: $(1 - 1/k)$-approximation for cardinality matching and vertex-weighted matching.
- Proposal making algorithms: stable matching and edge-weighted matching.
- Primal-dual linear programming: exact and approximation algorithms for edge-weighted matching, 3/2-, 2- and Δ-approximation algorithms for b-edge cover
- Randomized algorithms: Sinkhorn-Knopp approximation algorithms for cardinality matching, $2/3 - \epsilon$-approximation for edge-weighted matching.
- Scaling based algorithms and $1 - \epsilon$-approximation for edge-weighted matching.
- Reduction to matching for b-edge cover problems.

- **Applications of matching and edge cover**
  - Graph construction and sparsification for semi-supervised classification and adaptive anonymization of data.
  - Matchings and sparse matrix computations: the Dulmage-Mendelsohn decomposition of bipartite graphs and block triangular forms of sparse matrices. The strong Hall property and predicting the nonzero structure of sparse factors. The sparse range space basis problem and the sparse null space basis problem.
  - Network alignment via approximate edge-weighted matching.
  - Submodular matching and edge cover: load balancing parallel computations, word alignment in natural language processing, data summarization in machine learning, applications to proteomics, etc.

- **Additional Topics**
  - Parallel matching algorithms with logarithmic depth and linear work
  - Inapproximability results for graph coloring
  - Randomized algorithms and Luby's algorithm for maximal independent sets
  - Fill reduction in sparse matrix factorizations

Textbooks. Several of the topics we consider are not discussed in the textbook literature yet, and it is my intent to write a book on this topic for publication. Hence we will read a number of papers describing work on these problems from the past 10-15 years. Still, textbooks on approximation algorithms and matroids will be helpful to provide background and context, and some of them are listed below.

- David P. Williamson and David B. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, 2011.
- Vijay V. Vazirani, *Approximation Algorithms*, Springer, Second Printing, 2003.
- G. Ausiello, P. Crescenzi, G. Gambosi et al., *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*, Springer, 2003.
- Ding-Zhu Du, Ker-I Ko, and Xiaodong Hu, *Design and Analysis of Approximation Algorithms*, Springer, 2012.
- Dorit S, Hochbaum (ed.), *Approximation Algorithms for NP-hard problems*, PWS Publishing Company, Boston, MA, 1997.

- Jamex Oxley, *Matroid Theory*, Second edition, Oxford University Press, 2011.
- L. Lovasz and M. D. Plummer, *Matching theory*, North Holland, 1986.

Grading. Students will do regular Homework problems, which will include problems derived from lectures (completing arguments that are sketched out, applying techniques from lectures to new problems), and also implementations of algorithms. Participants will also be invited to create Homework problems! Students will also complete a semester-long project, which could involve implementation of a more challenging algorithm, or reading a paper that describes the state of the art on a topic related to the course and background material, and then preparing a report and presenting it in class.