Graph and Network Algorithms: Theory and Practice
CS 69000-GNA, 3 Credits, Fall 2011
Prof. Alex Pothen
List of Suggested Projects

This is a partial list of suggested topics for class projects. You can either provide a written report of one of these topics in class, or implement an algorithm from one of these topic areas and submit the code and results of running the code on some test graphs.

For a report, you could read the suggested chapter(s) or papers, understand the details of the paper, and provide a written report expressing the ideas of the paper in your own words. The presentation should be complete (the proofs of all key results should be provided), so that it should be easier to read and understand your report rather than the sources on which it is based. You should also provide a brief context for the paper, what contribution did it make to the state of the field when it was written, how has the paper been used, what further advances have been made in the field since the paper.

You are also welcome to teach the material in class (you will have one lecture for this purpose towards the end of the semester).

For the implementation of an algorithm, you should describe the algorithm (not code, but using the pseudo-code notation for algorithms we use in class), the computational environment (programming language, libraries used, optimizations, compiler, processor), data structures, space and run-time complexity of your implementation, test problems used, and results. If you are comparing two or more algorithms on a set of test problems, you should learn of reporting results using *performance profiles*. See Elizabeth Dolan and Jorge More, *Benchmarking optimization profiles with performance profiles*, Mathematical Programming, Series A, 91: 201-213, 2002.

- **Shortest paths** have been used to compute centrality measures in networks. There are several notions of centrality, i.e., how central a given vertex is in a network. The concept of betweenness centrality of a vertex v computes the ratio of the number of shortest paths between every pair of vertices in the graph that goes through v to the number of shortest paths between that pair of vertices. A linear space and quadratic time algorithm for computing between centrality was provided by Ulrik Brandes in the following paper, which was an improvement over existing algorithms.

  Ulrik Brandes, *Faster algorithm for betweenness centrality*, J. Mathematical Sociology, 25, 163-177, 2001.

- **Graph Models of Social and Information Networks**. The Erdos-Renyi random graph model has distributions of degrees and shortest path lengths that differ from models of community, information, and biological networks, which tend to have power-law degree distributions and have small average path lengths (small-world networks). Chapters 12 and 13 of the following textbook has an accessible discussion of these issues, and it uses a generating function description of random graph models with specified degree distributions.

Mark E. J. Newman, *Networks: An Introduction*, Oxford University Press, 2010.

Modeling how epidemics spread on social networks is another topic of importance to society at large. There are models such as SIR (Susceptible, Infected, Recovered), and extensions that consider the possibility of reinfection. Again, an accessible treatment is in Chapter 17 of the Newman textbook.

- **Colorings**. My research group has done extensive research on vertex coloring problems due to their applicability to Automatic Differentiation: distance-1 coloring (the usual notion of vertex coloring), distance-k coloring, star coloring, acyclic coloring, partial distance-2 coloring of bipartite graphs, etc. You can find a number of our papers on our web page for graph coloring: `www.cscapes.org/coloringpage`.

  There are still a number of outstanding questions that are appropriate for a small research project. E.g., every planar graph can be vertex-colored in at most four colors; every planar graph has an acyclc coloring with at most five colors; but for star coloring, we do not know the precise bound. It is between 10 and 20. A simpler problem is to consider this for simpler classes of graphs, such as interval graphs, chordal graphs, cographs, k-degenerate graphs, etc. See the web page, `www.math.uiuc.edu/~west/regs/starcolor.html` and the Wikipedia entries on acyclic and star colorings.

- **Matchings**. There is a vast literature on this topic, including several books. The following book is a comprehensive study from a discrete mathematicians's perspective:

  Laslo Lovasz and Michael Plummer, *Matching Theory*, North Holland, 1986.

  An important graph decomposition of non-bipartite graphs is the canonical Gallai-Edmonds decomposition, which is computed from a maximum matching in the graph. The computation of this decomposition can simplify many computational problems on the graph.

  There are also variations on matchings, such as semi-matchings (where the matching condition of each vertex having degree at most one in the subgraph of matched edges is enforced only on one side of a bipartite graph), or b-matchings, where the degree on the subgraph of matched edges is at most $b > 1$.

  For bipartite graphs, there has been much recent work on efficient implementations of augmenting path algorithms and the preflow-push algorithm. See the following reports:

  I. S. Duff, K. Kaya, and B. Ucar, Design, implementation and analysis of maximum transversal algorithms, Technical Report TR/PA/10/76, CERFACS, Toulouse, France, 2010.

  K. Kaya, J. Langguth, F. Manne and B. Ucar, *Experiments on push-relabel-based maximum cardinality matching algorithms for bipartite graphs*, Technical Report TR/PA/11/33, CERFACS, Toulouse, France, 2011.

- **Network Flows**. The book by R.K. Ahuja, T.L.Magnanti and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications* is the classic text in the area, and the place to begin to explore for new results.

- **Bioinformatics**. This is again an area where combinatorial algorithms abound. The first has a good collection of problems in this area solved by algorithmic techniques (a new edition is about to come out soon). The second book is more limited in scope, but is more recent.

  Srinivas Aluru (ed.), *Handbook of Computational Molecular Biology*, Chapman and Hall/CRC, 2005.

  Ion Mandoiu and Alexander Zelikovsky (eds.), *Bioinformatics Algorithms: Techniques and Applications*, Wiley, 2008.

- **Parallel graph computations**. How can we implement shortest path algorithms on practical parallel machines? The first two papers below suggest two algorithmic approaches, and the third paper provides an experimental evaluation.

  U. Meyer and P. Sanders, *Δ-stepping: A parallelizable shortest-path algorithm*, Journal of Algorithms, 49: 114-152, 2003.

  M. Thorup, *Undirected single-source shortest paths with positive integer weights in linear time*, Journal of the ACM, 46: 362-394, 1999.

  K. Madduri, D. A. Bader, J. W. Berry, J. R. Crobak, and B. A. Hendrickson, *Multithreaded algorithms for processing massive graphs*, Chapter 12 in David A. Bader (ed.), Petascale Computing: Algorithms and Applications, Chapman and Hall/CRC, 2007.

  The following chapter on graph algorithms on multicore machines has interesting insights although it was written several years ago:

  David Bader and Guojin Cong, *Efficient parallel algorithms for multicore and multiprocesors*, Chapter 26 in Handbook of Parallel Computing, S. Rajasekaran and J. Reif (eds.), Chapman and Hall/CRC, 2008.
  (See also Chapter 31.)

- **Approximation algorithms for graph problems**. The book by Williamson and Shmoys listed below contains a systematic discussion of approximation algorithms, and it introduces almost all of the known techniques for designing approximation algorithms. One particular algorithm that I could suggest for you to read and, explain or implement is the edge coloring algorithm due to Vizing in Section 2.7, I can also provide references to other recent edge coloring algorithms.

  The edge coloring problem is one of interest to me due to its applicability to implementing parallel algorithms efficiently on multicore computers. Talk to me if you are interested in exploring this problem.

Here is the list of books included in the syllabus for additional background reading.
Reinhard Diestel, *Graph Theory*, Springer, 2000.
J. A. Bondy and U.S.R. Murty, *Graph Theory with Applications*, Springer, 2008.
Douglas B. West, *Introduction to Graph Theory*, Second edition, Prentice Hall, 2001.

Alexander Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Volume A, Springer, 2003.

Recent books on networks and approximation algorithms include

M. E. J. Newman, *Networks: An Introduction*, Oxford University Press, 2010.
David Williamson and David Shmoys, *The Design of Approximation Algorithms*, 2011.