**INDR 520 Network Models and Optimization**

**Course Project**

In the course project you are expected to investigate a topic of interest to you that is related to the course topics. The goal of the project is to let you search, encounter and independently absorb new material and if willing, implement algorithms and perform computational tests, and present what you have done to share your knowledge with your classmates. Different projects will attain these goals in different proportions. You can work in groups of two.

There are three main ways to approach the project:

1. **Reading project.** Find a few interesting but challenging theoretical or computational papers on a coherent topic, and write up an exposition that synthesizes their ideas. You will be graded on how much clearer/more informative your exposition is than that of the original papers. While you need not give all proofs in full detail, the reader should come away with a good understanding of why it all works. Original demonstrative examples would be useful.
2. **Theoretical research.** Develop a more efficient or simpler solution method/algorithm to an interesting network flow or network optimization problem. While you will presumably need to do some background reading, there is no set lower bound on it so long as you are able to advance beyond what is known. This kind of project will be graded on the extent of the improvement on previous results.

**3- Implementation project.** By the end of the course we will have studied many algorithms in theoretical form (in terms of worst case run time), but their average behavior on different data sets can be quite surprising according to their implementation. Select a problem that interests you, and implement an algorithm and test it with different data sets. You may also compare several algorithms or different mathematical formulations. This kind of project will be graded on how well you explain the algorithm or the mathematical model you are implementing, what kind of interesting heuristics you have developed, if any, what interesting test cases you ran them against, and how well you present and interpret the results. Be cautious— some implementations can take a lot of time and can be quite challenging. Working in a team will be helpful. Give serious thought to the design of the test instances—ideally, they will come from real-world problems or be generated to carry the characteristics of such problems. Alternatively, they may be specially designed to test certain strong and weak aspects of the algorithm and to show its performance according to particular problem characteristics.

*Regardless of which type of project you select, the end result should be an* ***at most 10 page report*** *describing the problem, the algorithm or the mathematical model with an illustrative example and your computational results, and a 10-minute video presentation. The presentation quality and report style will be a factor in your grade.*

Some questions you may ask:

* **Is collaboration allowed?** Yes, even encouraged for implementation projects. You will be able to get more done in groups. Group size should be limited to at most 2.
* **What topic should I work on?** The best topic to pick is one you are interested in. Some of you are already involved in some research project; some thought may reveal an algorithmic component related to it. You can do background reading on the topic (reading project), develop a theoretical model of a related problem and devise a solution method (theoretical project) or take some existing algorithm and try it out (experimental project).

If you are not working on any research problem yet, try browsing through what we have covered in class as well as papers (see below for some examples) to identify an area that sounds like fun and beneficial. You are not limited to the topics we have covered in class. Any paper that uses algorithmic ideas related to networks and a theoretical framework to devise more efficient solutions will be fine. While presenting a paper, you may need to do further reading to cover the needed background.

**Where can I find papers?** There are numerous sources of papers on algorithms and network problems. There are journals such as *Networks,* *Operations Research, INFORMS Journal on Computing*, and *SIAM Journal on Computing* where you can find relevant and interesting articles. Also, some of the best work in the area is published in three refereed conferences each with yearly proceedings:

• The ACM Symposium on Theory of Computing (STOC).

• The IEEE Symposium on Foundations of Computer Science (FOCS).

• The ACM-SIAM Symposium on Discrete Algorithms (SODA).

For the below example problems, you can think about: What algorithms are available for this problem? Can you think of a new approach? Can you implement an existing algorithm or your own algorithm?

1. Finding maximum flow within a distance limit. In an input graph with edge lengths and flow upper bounds, send maximum flow from *s* to *t* but the paths used to send flow should not have length more than *R*.
2. Finding expected maximum flow that can be sent from *s* to *t*, when edges may fail with given probabilities independently. This is a flow problem in a network with unreliable arcs.
3. Finding an expected shortest path length from a node *s* to a node *t*, when edges may fail with given probabilities independently. This is a shortest path problem in a network with unreliable arcs.
4. Comparing alternative capacitated vehicle routing problem formulations, e.g. compare flow-based formulation to the vehicle indexed formulation, consider different approaches for subtour elimination. See: G. Laporte, What you should know about the vehicle routing problem, *Naval Research Logistics*, 54:8, pp. 811-819, 2007.

Some more possible topics can be listed as follows:

1. Multi-commodity network design problem; considerations of robustness, resilience and reliability when uncertainty exists in demand and capacity values
2. Time-dependent costs or travel times in shortest path and routing problems
3. Generalized vehicle routing problem: transformations and alternative formulations
4. Clustering in networks for partitioning routing problems
5. Hierarchical clustering algorithms in networks
6. Modeling vulnerability and disruptions in networks, random failures
7. Interdependencies in infrastructure networks, the impact of cascading failures
8. The minimum latency objective in routing problems
9. Prize collecting and selective routing problems
10. Online and adaptive algorithms in shortest path problems, e.g., the Canadian Traveler Problem
11. The impact of information updates and uncertainty in travel times in shortest path problems
12. Bicriteria shortest path algorithms
13. Multi-criteria shortest path algorithms
14. Generation of dissimilar and short alternative paths
15. Push and relabel algorithm implementation for the efficient solution of the maximum flow problem
16. Dynamic network optimization problems; problems in which demand is mobile and changes location over time; location of facilities or delivery to demand points on a network when demand is mobile