ECE 3484 - Combinatorial Optimization

Project 1: Develop a benchmark and evaluate exhaustive algorithms

Put together a set of instances for your problem that you will use to evaluate your algorithms. The largest instance should be at least 100 times larger than the smallest instance. Also, the structure of the instances themselves should vary. For graph-based problems, you should make sure that your instances have varying connectivities and node degrees. In all cases, make sure that some of your instances will be easily solvable and some will be very difficult to solve. Note that you will also need to define a uniform input format for these instances.

The first group of instances should be randomly generated (i.e. synthetic). Do research to see if anyone has previously discussed how to generate random instances of your problem. If there is no previous work on this, develop a technique that is convincing. Your goal is for your benchmark to explore the "instance-space" as uniformly as possible. This benchmark should contain at least 100 instances. Note that there has been a significant amount of work on generating random graphs.

The second group of instances should be those that have been considered by others. For most problems you will be able to find published hard instances that researchers have worked on. Often they are in papers or online. Put together a collection of published instances of varying sizes.

In your writeup, explain how these benchmarks were constructed and argue that these instances are representative of large class of possible instances and that it will be useful to evaluate your algorithms using them.

Develop and implement an exhaustive optimal algorithm for solving instances of your problem and evaluate it on the benchmarks. Try to find an ordered technique for considering all possible solutions to each instance. For example for the Travelling Salesman Problem you would consider every Hamiltonian cycle and select the cheapest. Try to ensure that every solution is considered exactly once. Apply the exhaustive algorithm to your benchmark with a 1 minute limit on CPU time and compute the fraction of the benchmark that has been solved to optimality. Then increase the CPU limit to 5 minutes. What are the characteristics of the instances that were solved? What size instances can be reliably expected to be solved quickly?