Greg Daniels
A00798340
Assignment 2 TSP

Algorithm Descriptions

All of the algorithms were implemented using an exhaustive depth first recursive method. On each call the algorithm would call itself for every other node not in the tour yet. Once it had used up all of the nodes it would check its cost against the best cost so far, if better it would save that path.

Algorithm b was as described above but it would check to see if its current path was longer than the best path, and if so would stop on that branch.

Algorithm C was also the same as the previous except at each step the order of the recursion was determined by the distance from the last point.

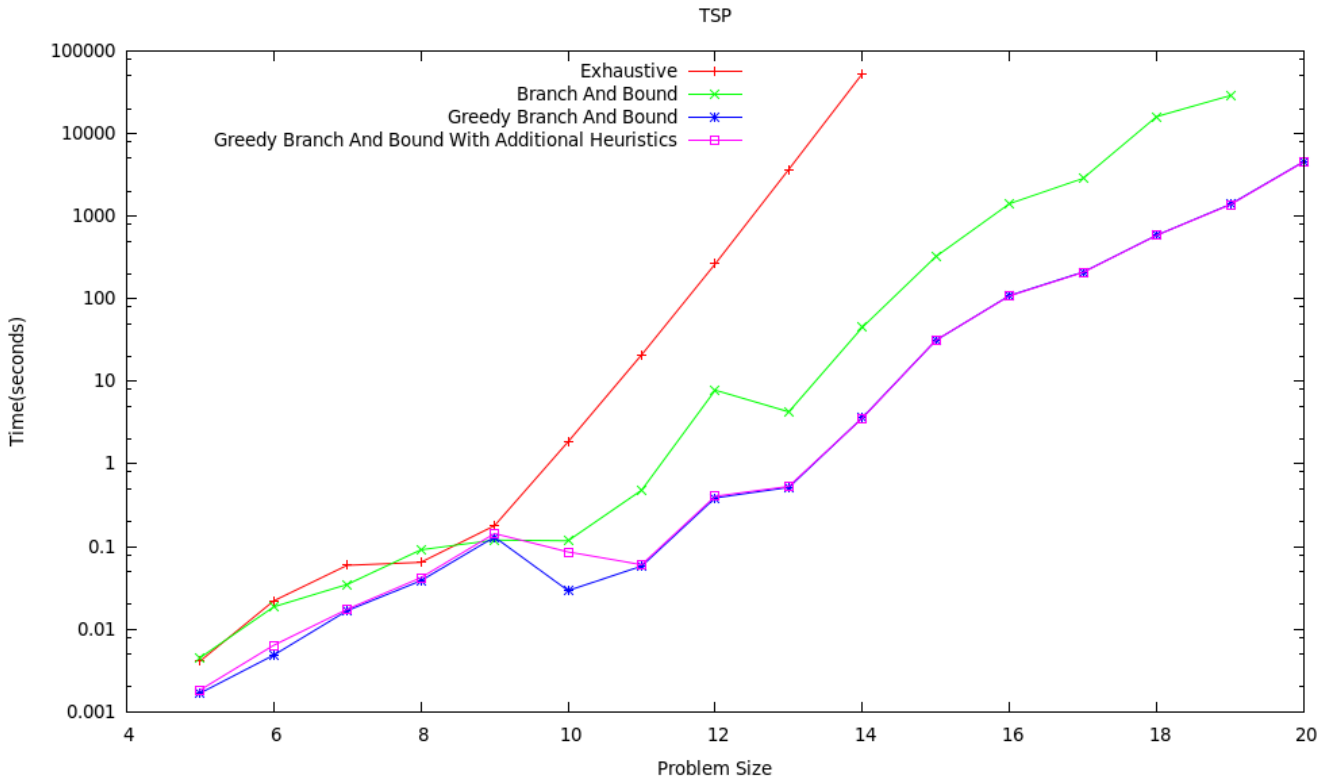
Algorithm D was the same as the previous with the addition of the heuristic that estimates the amount of work left to be done and uses that in addition with the current cost to cause the branch to fail earlier.

The algorithms were timed supposedly to the nanosecond but the nanosecond timing features in java are only accurate to the millisecond.

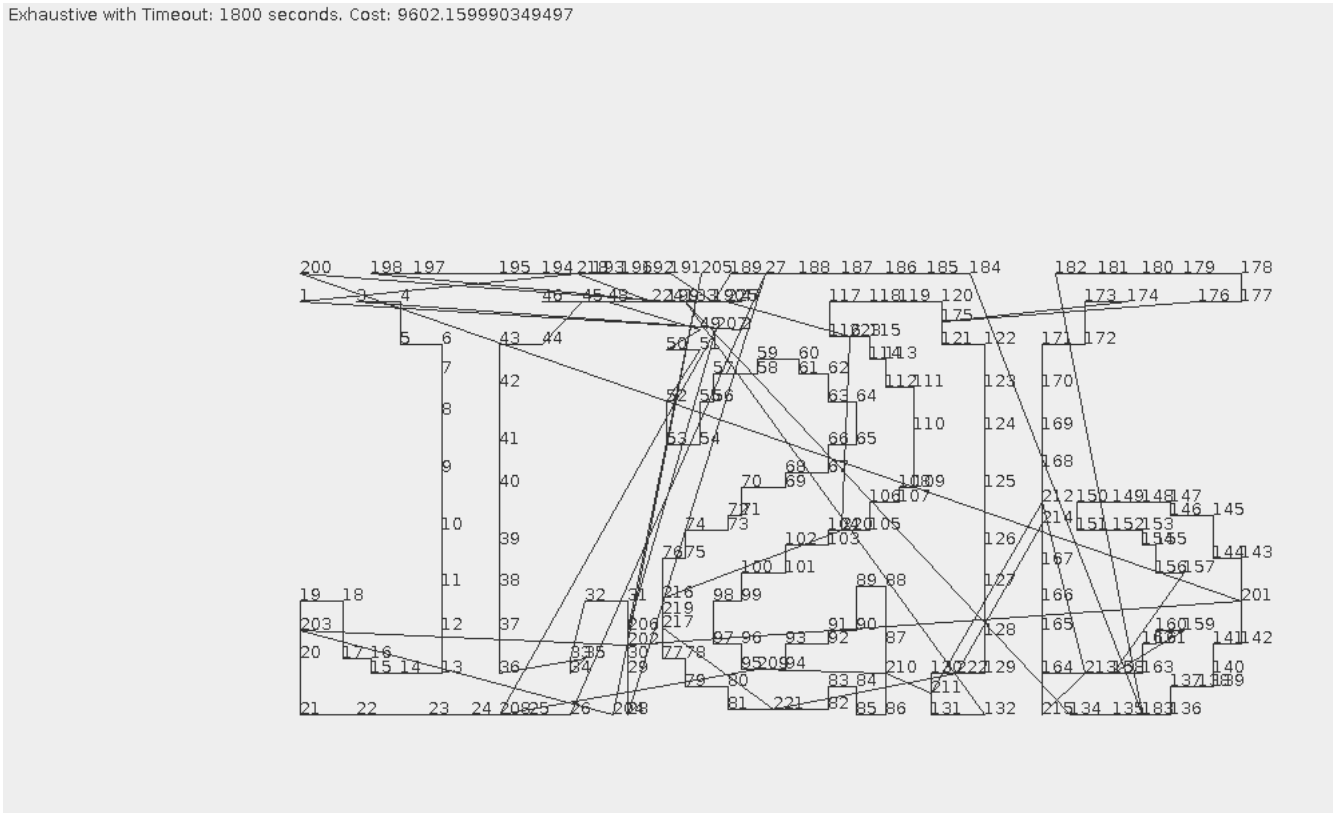
All of the algorithms were run on the same set of graphs, and allowed to run until it became clear that it would not solve the next larger size problem in a reasonable amount of time(meaning days). They were ran on graphs ranging from 5 nodes to 25 nodes. The problems were generated by randomly printing two floats out to a file as x and y coordinates.

As can be seen below all of the algorithms resulted in significant improvements, except for the heuristic one. The reason is that the estimate for how much path is left is very loose. The reason that the additional heuristics may not have added much performance is because of how loose bounding was.. So it remains very close to the greedy branch and bound.

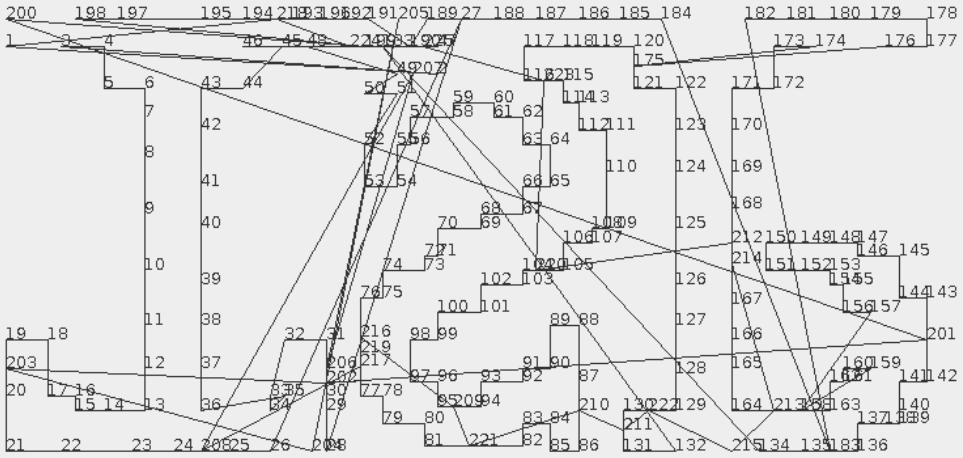
Data used to plot the graph is included in the gnuplot folder they are all text files without extensions. The cost and the path found are included along with them running time. The sample problems are found in the graphs folder, and the source code is included in the src folder.



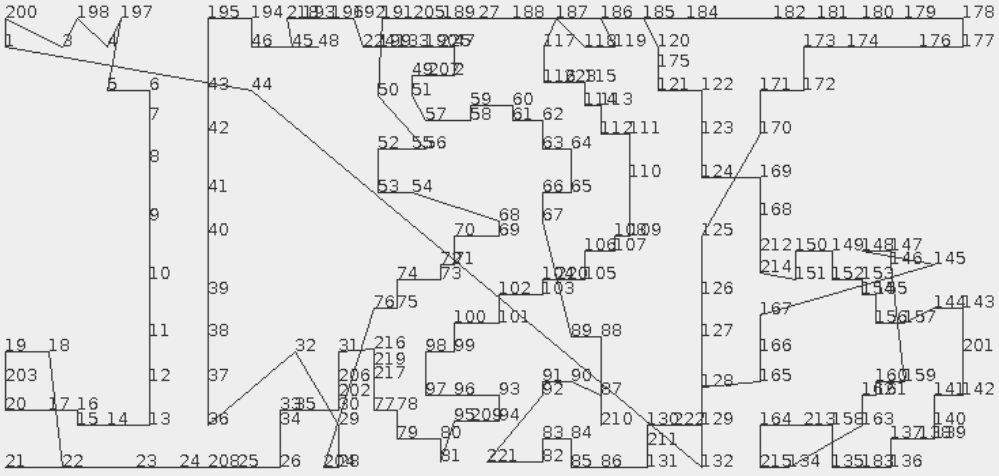
The next 4 images are the plots of the best paths found for the various algorithms on the provided problem. A time cut off of thirty minutes was used to produce these.



Branch and Bound Timeout: 1800 seconds. Cost: 9254.252247856924



Branch and Bound With Greedy First expansion Timeout: 1800 seconds. Cost: 4749.1030223070775



Branch and Bound With Greedy First expansion with aditional heuristics Timeout: 1800 seconds. Cost: 4749.1030223070775