CS 3510 Traveling Salesman Implementation
Samuel Shapiro & Kenedy Thorne

Introduction and Process

First, we attempted a simple greedy algorithm which attempted to find the shortest tour by choosing a random starting node and implementing the nearest-neighbor heuristic. However, this proved to produce paths typically over our 33,000 maximum tour length. Thus, we started looking into other algorithm implementations and eventually reverted to a greedy approach but improved it by adding the 2-opt heuristic. After playing around with the algorithm, we found it more time efficient to use a random generator to pick a starting node and have the rest of the initial path simply be in linear order rather than starting with the nearest-neighbor implementation. This approach was faster in our testing and would require much less setup time on very large sets of nodes.

Overall, we decided to utilize a linearly ordered path and implement the 2-OPT heuristic, which iterates through the path and measures whether or not switching the path between two nodes would shorten the overall tour length. If it does, it saves this new path as the best path and repeats until no more improvements can be found.

Breakdown of our Algorithm

1) First, the arguments are read in to get nodes into an input file.
2) The output file is cleared of content.
3) The time limit argument is read in and stored in a variable.
4) A start time is taken in order keeping track of the time for the duration of the algorithm.
5) The input file is read into a 2D array containing each node and its information.
6) A 2D array measuring of the distances between each node is created.
7) Then, a function is called to generate the initial path.
   a) This function picks 10 random starting points out of the nodes given and creates a tour by linearly following the rest of the nodes in the list provided.
   b) The distances of each tour are calculated by calling another function (findTourLength).
   c) The tour with the minimum tour length is chosen as the initial tour & is returned.
8) The 2-OPT heuristic is run on this selected tour, which searches through the nodes to find path optimizations by swapping the paths between the nodes and comparing adjusted tour lengths.
   a) Time is checked in each iteration to ensure that the limit has not been reached. If it has, the loop quits, saving the shortest path that has been generated so far.
   b) Otherwise, it continues to iterate through the best path, making changes until there are no improvements found. The best tour is then returned.
9) The distance of the final tour as well as the final tour are written to the output file.

Rationale

The reason we chose to have our initial path contain a random starting node and then the rest of the nodes in order is because we found that the initial path isn't as crucial as the implementation of the 2-OPT algorithm on that path. Very little time is spent on creating this type of initial tour, which allows most of the time to be spent on improving the selected tour. In nearest-neighbor or other situations in which a better initial tour is created, time can be wasted, especially with very large data sets. We found it less necessary to spend more time running a greedy algorithm to get the initial tour, especially since our testing showed longer overall times on a greedily created path than on a linear path.

However, we didn't want to have only one random tour that may end up being unnecessarily long. This is why we chose to create 10 random linear tours and choose one with shortest tour length in order to have a good starting point for the 2-OPT to then work on. This can be done efficiently and significantly decrease the initial tour length.

Results

Since we have a random component, we ran the practice test 10 times to get our average TSP cost:

28486
23 29 28 26 25 27 24 16 20 14 13 9 7 3 4 5 1 2 6 11 10 8 12 15 19 18 17 21 22 23

28215
27 25 26 28 29 21 23 22 18 19 15 12 8 10 11 6 2 1 5 4 3 7 9 13 14 17 20 16 24 27

29308
11 12 15 19 22 23 21 29 28 26 20 25 27 24 16 17 18 14 13 8 9 7 3 4 5 1 2 6 10 11

28765
22 23 21 29 28 26 20 25 27 24 16 17 18 14 13 9 7 3 4 8 5 1 2 6 10 11 12 15 19 22

28556
4 5 1 2 6 10 11 12 15 19 18 22 23 21 29 28 26 20 25 27 24 16 17 14 13 8 9 7 3 4

28067
21 17 18 19 15 12 8 10 11 6 2 1 5 4 3 7 9 13 14 16 24 27 25 20 26 28 29 23 22 21

28486
18 19 15 12 8 10 11 6 2 1 5 4 3 7 9 13 14 20 16 24 27 25 26 28 29 23 22 21 17 18

28494
12 15 19 18 22 23 21 29 28 26 20 25 27 24 16 17 14 13 9 7 3 4 5 1 2 6 11 10 8 12

28486
18 19 15 12 8 10 11 6 2 1 5 4 3 7 9 13 14 20 16 24 27 25 26 28 29 23 22 21 17 18

28215
28 29 21 23 22 18 19 15 12 8 10 11 6 2 1 5 4 3 7 9 13 14 17 20 16 24 27 25 26 28

Calculations

Average Cost of Tour = 28507.8
Standard Deviation = 345.8