

Assignment #5 – Traveling Salesman Problem

The “Traveling Salesman Problem” involves finding the shortest Hamiltonian path in a graph. A Hamiltonian path is a path that visits each node exactly once and returns to the starting vertex (http://en.wikipedia.org/wiki/Hamiltonian_path).

The implementation I used finds the shortest Hamiltonian path when given a set of European cities using the brute force method. This method will try every possible Hamiltonian Cycle and return the one that gives us the shortest path.

This implementation is a modified version of the program found here: <https://watchmaker.dev.java.net/examples/salesman.html>. The program will have you select a subset of the given European cities. It will then create a graph where the edge weights are the distances between cities. From there it will determine the optimal path.

As an example, say I choose the first 6 cities. The results will look like:

ROUTE: Amsterdam -> Copenhagen -> Berlin -> Athens -> Brussels -> Dublin -> Amsterdam
TOTAL DISTANCE: 6410.0km

Given a number of cities n , there will be $(n-1)!/2$ possible paths to evaluate. This is because we are given a starting city, which leaves us with $n-1$ choices for the second city, $n-2$ choices for the 3rd and so on. Since the direction we take our tour does not matter, we divide the total amount of tours by 2 (<http://www.tsp.gatech.edu/problem/pcb3cnt.html>).

The following table gives us the runtime to find the shortest Hamiltonian path and how many paths were evaluated according to the previous formula.

N	Time (Seconds)	Number of paths
7	0.0020	2,520
8	0.0090	20,160
9	0.0350	181,440
10	0.163	1,814,400
11	0.64	19,958,400
12	6.076	239,500,800
13	69.141	3,113,510,400
14	Undetermined	43,589,145,600