Traveling Salesman Problem

Introduction

We researched graph theory and the traveling salesman problem during the third week of Governor's School. Through this project we learned many different methods of using graph theory to get to the most efficient answer.

Background

The travelling salesman problem asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" There are many different options to solve this computational complexity. We could solve it using brute force, finding every possible route and calculating the shortest one, but that would take over 70 million centuries for just twenty cities! Instead, you can use algorithms, such as the nearest neighbor algorithm that we chose, to maximize efficiency. Another option is to use an already-existing algorithm to check your work.

Three Members Presenting Our Project

Main Content

We used two different methods in a project to simplify and exhibit the Traveling Salesman Issue. We took the different high schools of everyone in our class and researched to find the best way to visit all the schools in a round trip. The two methods we compared were the nearest neighbor method and the foreign software Mapquest. The following describes the nearest neighbor method that we graphed on the poster with the map.

•**Start** :anywhere on the graph

•**First Step**: Move from starting school to the next closest school, the nearest neighbor.

•**Middle Steps**: From each school go to its nearest neighbor, *choosing only the school that hasn't been visited*.

•**Last Step:** Once all schools have been visited, return to the starting one.

References

•https://en.wikipedia.org/wiki/Travelling_salesman_problem

•**Our Teaching Assistant Lorna Treffert**

• **Mapquest.com**

Conclusion

Through the process of physically mapping out each method, we learned that the nearest neighbor method worked very well until the end, when you had to take a long, useless path back to the beginning. Our online algorithm that we discovered revealed that you should almost look at your path as a circle and aim to follow that shape of path.

THE UNIVERSITY OF TENNESSEE KNOXVILLE

Gracee DeJarnette, Ashley Hamme, Hamid Djouadi, Nick Jones