Project Title: "The Traveling Salesman Problem"

New Mexico
Supercomputing Challenge
Final Report
April 3, 2019

Team Number: 6

School Name: Capital High School

Team Members:

Ariana Garcia

Arath Oscar Flores

Carlos Lopez

Melissa Gill

Sponsoring Teacher:

Irina Cislaru

Project Mentor(s):

Barbara Teterycz

Matthew Curry

David Gunter

Table of Contents

Executive Summary

The Traveling Salesman Problem (TSP) is a very commonly known mathematical problem that asks for the most efficient route possible with a given set of points (cities) that must all be visited [2]. The traveling salesman problem is very important because this problem is a general optimization, which means that its solution can be applied to many fields like transportation and in science [1].

There are many different algorithms that have been created specifically for this problem; we are using two algorithms called the "Nearest Neighbor" (NN) algorithm and the "Random" algorithm. The NN algorithm is a simple algorithm that works by choosing the next closest city possible, many which refer to as a "lazy" algorithm since it doesn't do much [13]. The random algorithm chooses a random path every time it is run, coming up with a new solution every different time. However, with the test that my team and I did, we found out that the NN algorithm works almost 73% more efficient than the random algorithm.

Over the course of this project, we got to expand our knowledge on coding, since most of us are new to the subject and are very passionate about it. We manage to understand how the TSP's most common algorithms work and how they can be applied to solve many different types of problems. This helped us get a better understanding of how we can experiment with other projects. Thanks to this project, we can be prepared for future coding classes and know what to expect of them thanks to the challenge and our amazing mentors.

Introduction

Our world is always innovating, looking for the most efficient and fastest solutions to our modern problems in life. The Traveling Salesman Problem has puzzled many people (including scientist) around the world for many centuries now. Its origin is not well known but more of the history of this puzzling problem can be found in the “background” section on page 5. This problem is truly perplexing because of the fact that there are so many solutions that could work but some tend to not always work like the random algorithm compared to others like the NN and Recursive algorithm, which will be later described in the “algorithm” section.

Thanks to our modern technology, this problem has been expanded more than what anyone could have thought of. There are many dedicated scientists who have spent most of their time on this problem and have found many different types of solutions using various methods. Many which are really close to a solution with different types of scientific experiments like ants and amoeba!

The goal of this project was to experiment with different types of computer algorithms and how efficient they really are. For example, in our project, we tested the NN and the random algorithm to see which algorithm would work best. The Traveling Salesman Problem has helped us expand our knowledge in the algorithms used to solve this problem. We manage to find algorithms that worked better than others but of course, it would have been very time consuming to test most of them. That is why we only tested two

that are very popular. The TSP is indeed a very essential problem, belonging to the computational mathematics section, that has been studied over many years now.

Description

Our project is based on the Traveling Salesman Problem which many refer to as the TSP as a short name. The TSP, as mentioned before, tries to accomplish its goal of visiting a group of cities in the most efficient way possible. It has been a puzzling problem for many, but we have managed to solve it using two different kinds of algorithms. Many of whom have used different sources other than computers.

For this supercomputing challenge, we decided to use two algorithms called the NN and random algorithm. As previously mentioned in the "Executive Summary" section, the NN algorithm work by choosing the next closest city there is. This algorithm is, in fact, better than our random algorithm which we will describe more in the "Algorithm" section. We ran both algorithms in NetLogo just because we know that NetLogo illustrates more of what is coded. We coded the United States and applied the two types of algorithms and run them

Although the term "traveling salesman" is not used anymore, this is still a very important problem because it applies to all kinds of transportation and even other problems. The algorithms used to solve this problem can be applied to other problems. This is such an important problem because of the many ways one can solve it and the outcomes of those solutions. They can help us understand our world better. These algorithms can be used by the people who work for the post office since they need to send mail fast and make it less expensive. Applications such as "Google Maps" uses algorithms that help determine the most efficient trajectory one can take, which is very convenient for

people that travel a lot. The Traveling Salesman Problem is not just a computational/mathematical problem studied by scientists, its outcome can help find new solutions to our modern day problems.

Background

The TSP is indeed a very complex problem that has been solved with many different types of algorithms and experiments, just as previously mentioned in the introduction. When we were evaluated in February, we got feedback from our judges (Drew Einhorn, Howard Delacruz- Bancroft, and Jeri Roberts) about how scientist are looking to solve the TSP with ants and amoeba. At Keio University, researches assigned an amoeba to solve the Traveling Salesman Problem, in which the amoeba found" nearly optical solutions" [12].

The exact origins of this troubling problem remain unclear, but its earliest example can be traced down to a German handbook called *Der Handlungsreisenden - Voyageur.* Published in 1832, this book did not give a mathematical solution or equation of the problem but it did give a precise description of what the problem was about [3]. One of the things he cites are his ideas on what the traveling salesman can do in order to obtain success, but since this is part of the original book the page found will be in its original language which I believe is German or Hungarian[4]. We also know that the TSP was created by mathematicians whose names were W.T. Hamilton and Thomas Kirkman, but the first mathematical expression of the TSP was made in 1930 by the American mathematician Karl Menger [5].

While doing research, we found out that in previous years another supercomputing team had done a similar project involving the TSP. The student, named Ian Wesselkamper, mentioned in his final report, he mentioned how he solved the problem by using the “brute force” algorithm and comparing it to the greedy algorithm (also known as the Nearest Neighbor algorithm).

Wesselkamper stated in his calculations that the brute force algorithm was much efficient than the greedy algorithm due to the fact that it “was inferior and would too often find wrong answers to more complicated random generated cities sets”. As a team, we agree with Wesselkamper do to the fact that if it chose the nearest cities and if the last one was all the way across it would have to travel a long distance. He proved that the TSP is “solvable in non-polynomial time” but was not sure it was the most effective solution [14]. We initially wanted to experiment with the “Recursive” algorithm, or what Wesselkamper referred to as the “brute force” algorithm. This algorithm works best because it gives an exact solution to this problem. This algorithm will be further explained in the ”Algorithm” section.

Algorithms

Our project is made up of different algorithms that output different solutions. One of the algorithms that we used was the “Nearest neighbor algorithm. The NN algorithm chooses the next closest city to the “salesman” and keeps choosing the next city until it has visited all the cities. This is a very simple algorithm that has been used to solve the TSP

with a 25% optical solution. Which is better than the other algorithm we decided to test because its percentage is higher and has more chance of solving it.

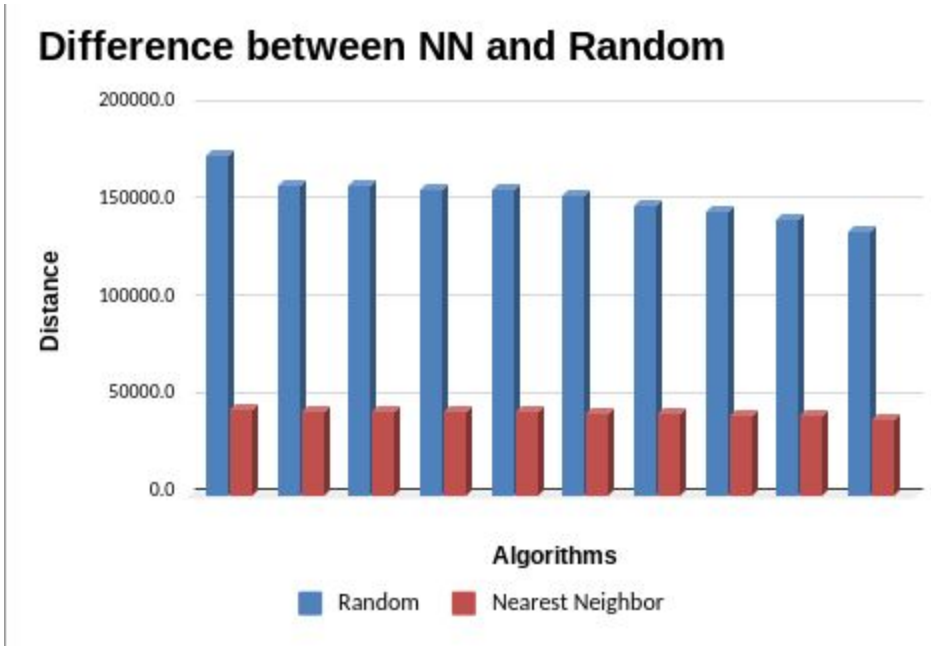
The other algorithm that we decided to experiment with is called the Random algorithm. Like the name says, this algorithm will choose a random path of cities to follow, which is not a great approach to the TSP. The random algorithm can choose any random path and follow it but it will not give the same results as the NN algorithm because it is always something different.

Since this was the first time for everybody as a member of the challenge there are a lot of things that we did not get to accomplish as we would have liked in the beginning. One of them being the recursive algorithm. The recursive algorithm calls itself with "simpler input values" [8]. This algorithm simply calculates the total distance for every possible route and then proceeds to select the shortest one [10]. This algorithm works better than the NN and random algorithm because of the simpler way it showed the output and exact solution. Even though the NN algorithm is better than the random algorithm it is still flawed just like Wesselkamper mentioned in his final report. This is due to the fact, that in some cases the NN algorithm might get tricked because of the fact it chooses the next closest city and it might get to a point where the next closest city is all the way across. That length it will have to travel will add to the extra distance and will not really work.

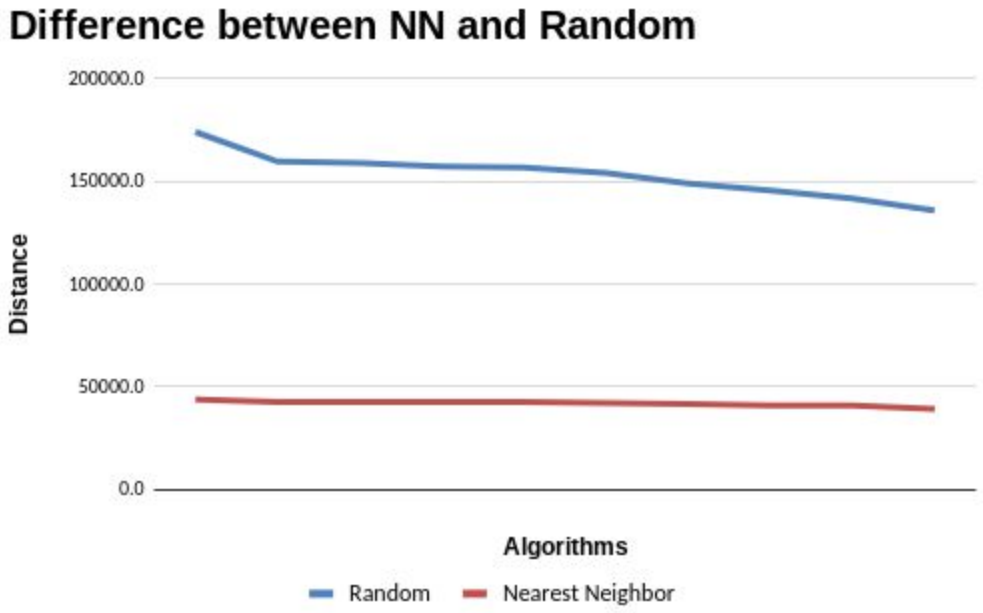
Results

We decided to test the NN and the Random algorithm in a NetLogo environment because we believe it would illustrate, in a more detailed way, how the algorithms differ from each other. They are two of the most well-known algorithms used for this problem, but the NN algorithm works better than the random algorithm.

Through a series of tests, we found out that the NN algorithm if almost 73% more efficient than the random algorithm, which can be seen on the chart down.



This chart is the same as below but illustrated differently to see the maximum difference between both algorithms.



We decided to run both algorithms ten times to get a good average of how well the algorithms work. This is the data we got from both performances. The percentages mean how much better each run was than the last one. They also state the difference between the random and the nearest neighbor algorithm. As the test went along, the percentage number was higher, which is good because that means that the NN algorithm performs best than the random one.

Random	Nearest Neighbor	Percentage
173973.3	43750.8	25.1%
159490.2	42619.9	26.7%
158957.3	42584.5	26.8%
157239.9	42584.5	27.1%
156570.7	42561.0	27.2%
154043.4	41971.0	27.2%
148993.3	41582.1	27.9%
145441.5	40915.7	28.1%
141499.8	40899.6	28.9%
135857.3	39236.9	28.9%
	Avg	27.4%
	Difference	72.6%

We also decided to include the average difference between the two and the total difference of how well the NN algorithm is compared to random.

Conclusion

The Traveling Salesman Problem has come a long way and in the process; its solutions have evolved and also its methods, which has helped many scientists use it in many other ways. In our project, we decided to test both Nearest Neighborhood and Random algorithm to show the difference between them. In the early stages of this project, we decided that the NN algorithm would work better than the random algorithm. We knew that the NN algorithm would work better because it chooses the closest city it can find saving the extra distance it would have to travel compared with the random algorithm.

After running the test, we discover that our calculations were correct. We solved the Traveling Salesman Problem by using the Nearest algorithm and the random algorithm. With the tests that we did, we found out that the Nearest algorithm is almost 73% percent more efficient than the random algorithm. The NN algorithm is 72.6% more efficient to be exact, which proved our point that this algorithm would be more efficient than the random algorithm.

Getting to show that we were right was very significant to us because this was our first coding project. We were amazed to be able to get help from our mentors and to get to solve one of the most commonly known mathematical and computational problem there is by using the NN and random algorithm.

Recommendations

Results are not always what one expects at the beginning of a project. Things change over the course of its period and there are many things that we, as a team, did not get to fulfill.

If we would have had enough time we would have liked to have tested out the recursive algorithm because it is an exact solution to this troubling problem. It would be a great experience to be able to test this algorithm and compare it to the other two algorithms. After testing the recursive algorithm we could be able to show and compare it with the data that we got from the other two algorithms that we managed to test.

Another thing we would have wished to have done is to have done more research on the amoeba and ants solution that was mentioned to us by our judges from February just because it sounds like such an interesting topic that we believe will intrigue a lot of people. We also were planning on translating our code from NetLogo to Python, but due to the time limit, we had to move forward without it. We think that it would be a good idea to do it because in the process you work with different mentors and learn two coding languages. We believe that the Traveling Salesman Problem is a very intriguing computer/mathematical problem. We leave the TSP in the hands of future supercomputing teams that will also be curious about the Traveling Salesman Problem and get to test things that we didn't get time to do and keep moving forward with this classic algorithmic problem.

Acknowledgements

As a team, we would like to thank the following people for mentoring us and assisting us in every way possible:

Mark Galassi: Who informed us about the Traveling Salesman Problem and helped us get started on our project.

Barbara Teterycz: For being every step of the way with us and supporting us in every which way possible. With mentors and NetLogo classes that helped us throughout the challenge.

Matthew Curry: For mentoring us in the TSP and its algorithms, for commuting all the way from Albuquerque to Santa Fe once a week in order to work with us, and just for being there we found ourselves in doubt.

David Gunter: Who helped us translate our NetLogo code to Python and do research about the different types of experiments scientist have conducted in order to solve the TSP.

Irina Cislaru: For always keeping us on our feet and keeping us updated throughout the challenge, assisting us in every way possible, and of course being our teacher and sponsor.

Ian Wesselkamper: Who gave us permission to cite his final report and use it as a resource.

Citations

[1] http://www.math.uwaterloo.ca/tsp/

[2] https://www.mathematics.pitt.edu/sites/default/files/TSP.pdf

[3] http://www.math.uwaterloo.ca/tsp/d15sol/dhistory.html

[4] https://zs.thulb.uni-jena.de/receive/jportal_jparticle_00248075

[5] mysc.altervista.org/the-travelling-salesman-problem/

[6] https://cameroncounts.wordpress.com/2012/07/19/the-traveling-salesman-problem-an-optimization-model/

[7] https://www.geeksforgeeks.org/greedy-algorithms/

[8] https://www.cs.odu.edu/~toida/nerzic/content/recursive_alg/rec_alg.html

[9] https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/

[10] https://guide.freecodecamp.org/algorithms/brute-force-algorithms/

[11] https://medium.com/basecs/speeding-up-the-traveling-salesman-using-dynamic-programming-b76d7552e8dd

[12] https://phys.org/news/2018-12-amoeba-approximate-solutions-np-hard-problem.html

[13]

https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7

[14] http://www.supercomputingchallenge.org/13-14/finalreports/82.pdf

[15]

https://medium.com/basecs/the-trials-and-tribulations-of-the-traveling-salesman-56048d6709d