**Real Time Delivery Problem**

Research Paper on Delivery Problem and its Solution Using Greedy Algorithm

*Himanshu Kaushik, Archit Nahta, Srinivasan Sharma, Anurodh Acharya*

*Students (B.tech II Semester) Department of Computer science*

*VIT, Vellore-632014*

I. Abstract

As we know that, Now-a-days everything is getting connected to internet. Internet on one hand has established a connection between customers and products available in stores far from them.

The real time delivery problem works based on the maps and graphs. Basically, the location for any delivery in present days scenario is taken on the basis of geometrical coordinates, ie- earth’s longitude and latitude. Eg: (32.26, 36.24).

Travelling salesman problem(TSP) consists of number of cities and salesperson. The salesperson needs to travel each and every cities which is starting from a one place and returns to the same city at the end of the delivery trip

One of the main goals of this project is to improve the performance of our already existing delivery algorithms. Generally, the best approach for improving the performance of a branch and bound algorithm is to improve the lower bound. Our new strategy is to use a greedy method for edge selection, rather than trying all possible path permutations.

The values in the rows and columns in the matrix are made to the largest distance we assume (i.e. 10,000) during each iteration. Therefore, we try to determine the edge that would provide us with minimum cost in that particular iteration.

Once we reach from one location to another, we find the nearest neighbor from that location. That neighbor is visited next. Then this process continues iteratively till every address is served. After serving every location, delivery person returns to the starting addresses. Meanwhile, at every visit, we add the cost to a total and locations numbers to a list after every iteration.

II. Introduction

Real time delivery problem is one of the most vital method in the application of transportation service. The world needs a better way to travel, in particular it should be easy plan to an optimal route through multiple destinations. It can be elaborated as a delivery man who travels through all the addresses in a city designated by the shortest distances, where each address can only be traversed once. Solution of the delivery problem is the path traversed by this delivery man. At last, the best solution of this problem is the shortest distance path or path with minimum of travel routes. The Travelling Salesman Problem (called TSP) is a classic algorithm problem in the field of computer science. It is an algorithmic problem for finding shortest route between a set of given points and locations that must be visited exactly once. A depiction of the term ‘TSP’ is, a salesman who have to travel ‘n’ cities or addresses where, he must visit each city or address once. In the end, after travelling total distance, he must return to his base city from where he started to travel. This type of delivery system can be seen regularly now-a-days. Zomato, Swiggy, Dominos and many more online shopping websites are some real life examples of this delivery problem. Our main project goal is to apply a better TSP algorithm to solve real world problems.

The travelling salesman problem(TSP) can be explained in the following ways:

TSP={(T,x,t):T=(V,E) i.e. a graph}

x is function V\*V-> Z,

t belongs to Z

T is a graph that consists of salesperson’s tour with the cost

III. Literature Survey

Real time delivery problem comes with the problem of cost efficiency. Manually, we can find the shortest path between a maximum of 8-10 locations. But electronic computation is required for more, say for 100 addresses , we have 100! ways to travel all the addresses.

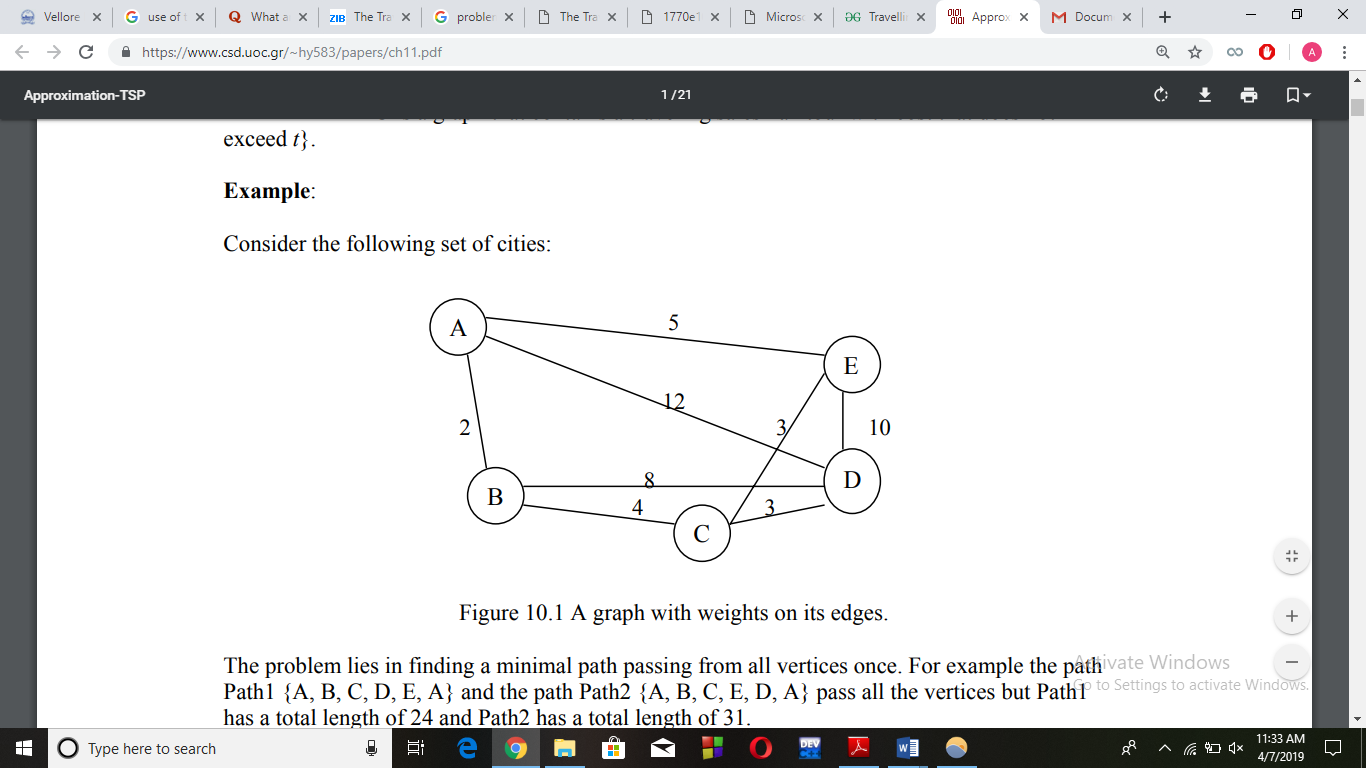
One of the main goals of this project is to improve the performance of our existing TSP algorithm. Generally, the best approach for improving the performance is improvising time to reach any node in computation. Our new strategy is to use a greedy method for edge selection, rather than trying all possible path permutations.

The major technical hurdle associated with this project is correctly implementing our new algorithm and improving the performance of previous implementation by a significant amount. In contrast with our original approach, where we would simply permute all possible solutions, the new algorithm is more selective in the paths it considers. Also, we no longer use a recursive function for calculating a particular path locally.

IV. Problem definition

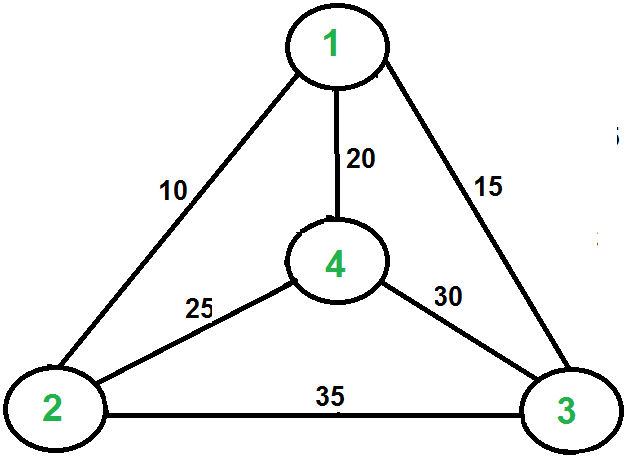
If the number of cities given are ‘n’, we need to find the distance between them. A tour has to be figured which visits each and every location exactly once such the sum of the travel is minimum.

Let us consider the given cities are alloted for the delivery problem

\

The problem is to find the minimum path which is passing from all the vertices once. In the given graph, the path A,B,C,D,E,A and the path2 A,B,C,E,D,A passes all the vertex whereas path1 is having the length of 24 and path2 has the length of 31.

Another Example:



Consider the given graph with 4 locations.

Starting from 1, choices to visit are: 2,3,4. Min= 1->2 =10

From 2, choices to visit are : 3,4.

Min = 2->4 = 25

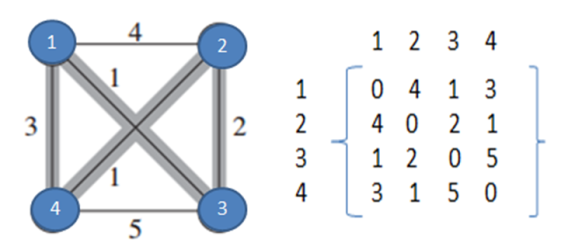
From 4, choice is 3

Min= 4->3 =30

From 3, Choice is 1 =15

Total= 10+25+30+15 = 80

Consider this next:



The path traced is

1->3 = 1

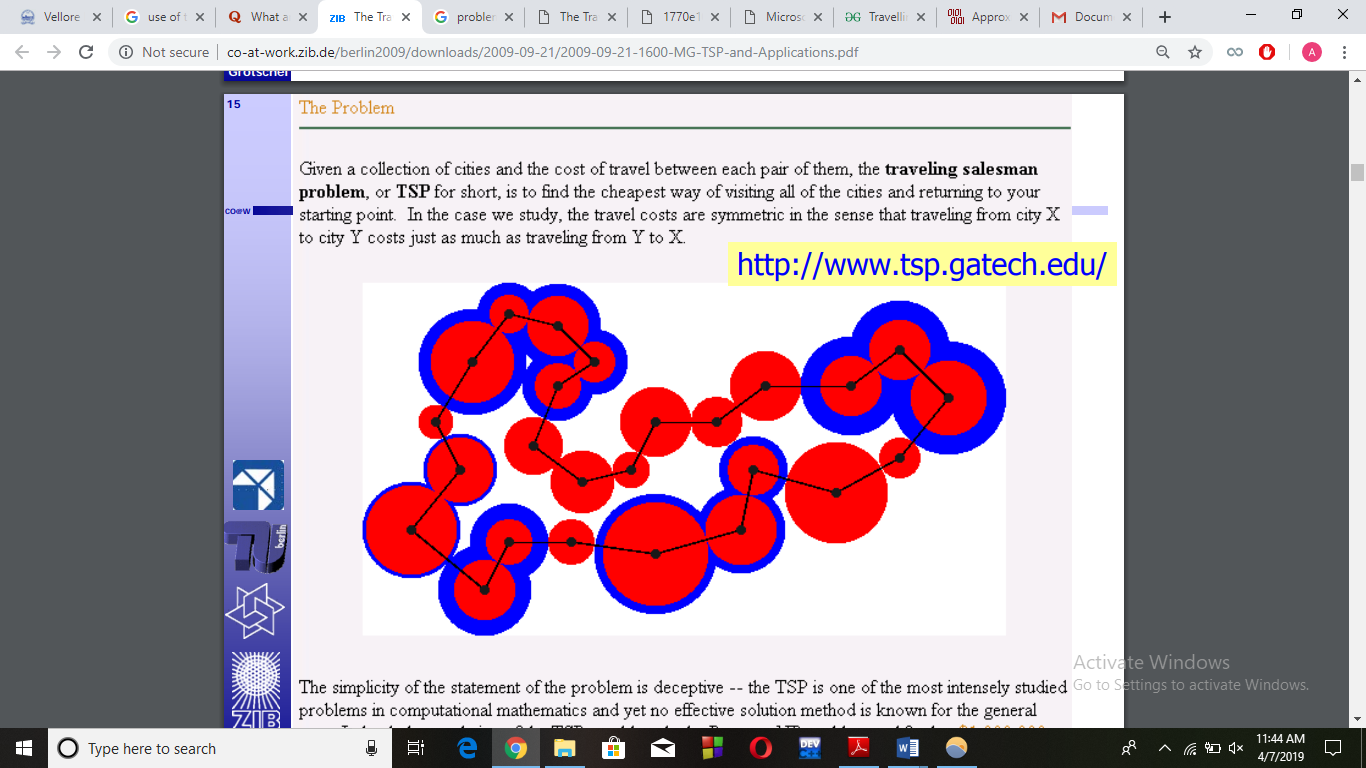
3->2 = 1+2

2->4 = 1+2+1

4->1 = 1+2+1+3

Total Cost=7

Consider the set of the given cities along with the cost of the entire travel between each and every pair, the algorithm finds the shortest distance for the complete delivery path.



The travelling salesperson person is among the most studied problems in the mathematics and still no effective solution has been figured out.

The resolution of TSP could settle the P vs NP problem that could transform the way of minimum cost travelling.

Already there is an ongoing research on whether P==NP. Whether the optimization problems can have a perfect solution computed.

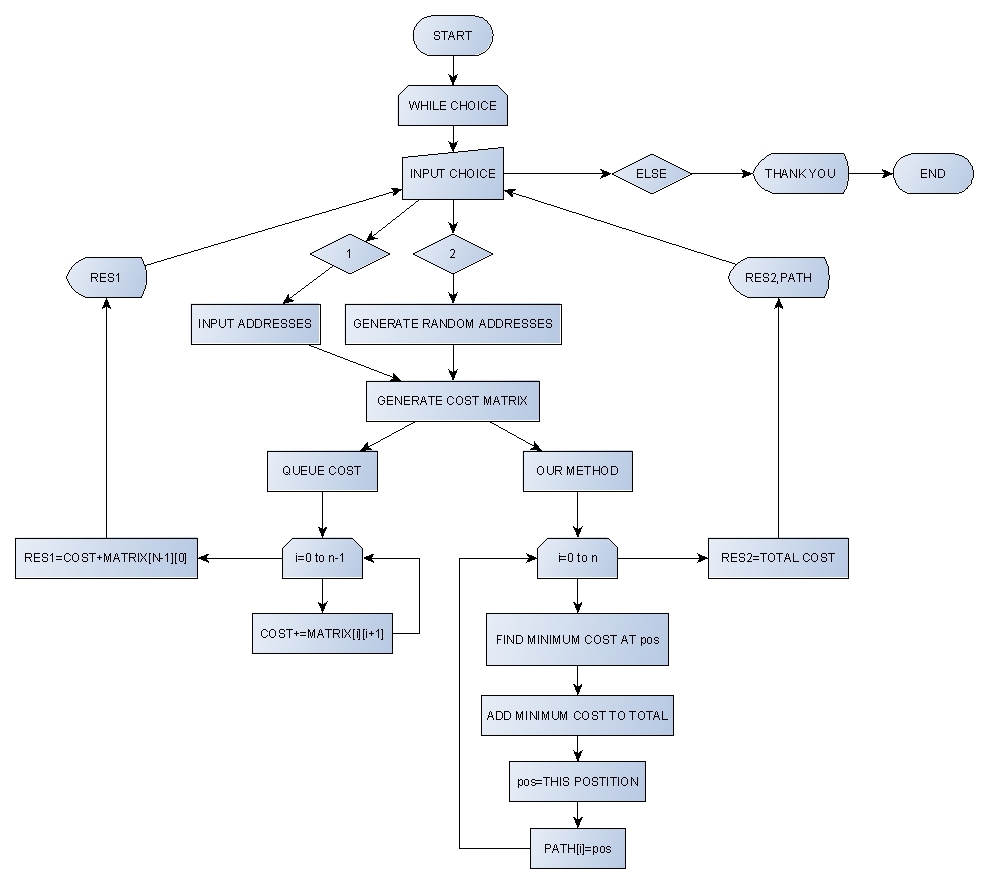
V. Concepts Used:

**Language:** C/C++

Functions: To be called for the different cost calculations and path determination.

Array: Used to store path details and cost matrix.

Control Flow: Manipulations of seek positions every time to go to desired position.



VI. Time Complexity:

The other ways to find the optimal solution are Branch and Bound, Dynamic Programming etc.

Our method has an advantage over them because of its lesser time complexity. That is O(n2)

Since,

Finding the minimum cost in from one location to another takes a complexity of O(n).

Finding the minimum cost for locations from n location to another n locations takes O(n\*n) unit of time.

Whereas,

Branch & Bound has a time complexity of O( (n-1)! ).

Dynamic Programming has a time complexity of O(n2\* 2n).

VII. Comparison:

|  |  |  |  |
| --- | --- | --- | --- |
| S No. | Number of Addresses | Queue Cost | Our Method’s Cost |
| 1 | 1 | 0 | 0 |
| 2 | 5 | 313.474670 | 289.051941 |
| 3 | 10 | 3219.107910 | 2896.343506 |
| 4 | 20 | 7740.936035 | 4435.609375 |
| 5 | 50 | 44568.17968 | 27758.699219 |
| 6 | 100 | Float Limit Exceeded | 40601.699219 |

When range of coordinates of delivery address is reduced:

|  |  |  |  |
| --- | --- | --- | --- |
| S No. | Number of Addresses | Queue Cost | Our Method’s Cost |
| 1 | 1 | 0 | 0 |
| 2 | 10 | 13.4867 | 12.274157 |
| 3 | 25 | 161.116 | 102.276726 |
| 4 | 50 | 280.677 | 124.393112 |
| 5 | 75 | 581.483 | 284.161163 |
| 6 | 100 | 500.201 | 124.585869 |
| 7 | 150 | 803.706 | 372.977966 |
| 8 | 200 | 1921.47 | 474.072662 |

VIII. Conclusion

We have observed the example for the given problem for the Travelling Salesman Problem which produces better result than the present famous algorithms. This is true for general and requires more research in the coming future.

We found the above results as expected. In the comparison section we can see how our algorithm overcomes the existing results.

This greedy algorithm for delivery problem proves to be more efficient in terms of time and space complexity.

After the research, we can observe that Delivery Problem is applicable today and will continue to be relevant in the upcoming days.

IX. References

1. Introduction to algorithms- Thomas H Cormen

2.

<https://simple.wikipedia.org/wiki/Travelling_salesman_problem>

3.

<https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/>

4. <https://www.thecrazyprogrammer.com/2017/05/travelling-salesman-problem.html>

5. <http://www.academia.edu/Documents/in/Travelling_Salesman_Problem>

6. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0201868>

7. <https://optimization.mccormick.northwestern.edu/index.php/Traveling_salesman_problems>

8. <https://nptel.ac.in/courses/111105039/20>

9. <https://www.youtube.com/watch?v=1FEP_sNb62k>

10. <https://www.youtube.com/watch?v=XaXsJJh-Q5Y>

11. <https://www.youtube.com/watch?v=e2cF8a5aAhE>

THANK YOU

X. Authors:

Himanshu Kaushik

-18BCE0579

Archit Nahta

-18BCE0584

Srinivasan Sharma

-18BCE0931

Anurodh Acharya

-18BCE2385

Students at VIT Vellore (2018-22) expected.