IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)
ISSN(E): 2321-8843; ISSN(P): 2347-4599
Vol. 2, Issue 6, Jun 2014, 59-72

A COMPARATIVE STUDY OF BRUTE FORCE METHOD, NEAREST NEIGHBOUR AND GREEDY ALGORITHMS TO SOLVE THE TRAVELLING SALESMAN PROBLEM

ANTIMA SAHALOT & SAPNA SHRIMALI

Department of Mathematics, Pacific Academy of Higher Education & Research University,
Udaipur, Rajasthan, India

ABSTRACT

A salesperson desires to travel from a city to all the other cities exactly once a time to sell his products and return back to the city where he started from. He wants to cover this in minimum time with minimum distance while travelling. This paper discuss on a proper path of travelling Salesman. TSP is an application of graph theory. In the term of graph theory, given a list of cities and their pair-wise distance, the task is to find a shortest possible tour (Hamiltonian Cycle) that visits every city exactly once. Even though TSP is an easy to understand, but it is very difficult to solve. Researchers have proven that Travelling Salesman Problem is NP-complete.

This paper explains a simple demonstration of the uses of modern graph theory by formulating a "plan of travel" between five metro cities: Delhi, Kolkata, Mumbai, Sheila, and Ahmadabad. The procedure included the utilization of the four major algorithms in graph theory to create a "plan of travel" that would model the more sophisticated technology used by computer system like map quest. This paper focused on finding the route with the lowest cost and the route with smallest distance. We also study the history and growth of TSP by graph theory.

KEYWORDS: Graph Theory, Minimum Distance, Minimum Cost, Route, Travelling Salesman

INTRODUCTION

Graph Theory is the study of graphs. Graphs are mathematical structures used to model pair-wise relations between objects. Graph theory can be used to solve problems in branches of Mathematics, Computer Science, and other scientific areas. We can make model an enormous number of real world systems and phenomena using graphs.[1, 5, 6, 17] There are many different problems in graph theory that have attracted much attention. One such problem is the Travelling Salesman Problem (TSP), which refers to a salesman who wants to find a shortest possible tour that visits every city exactly once and returns back to the city from which he started.[2, 3, 4]

In terms of graph theory, given a list of nodes (cities) and their pair-wise distance, the task is to find the shortest possible tour that visits every node exactly once. Since the start node and the end nodes are the same the tour creates a Hamiltonian cycle, which is a cycle in the graph that visits each vertex exactly once.[7, 8, 9, 10]

Mathematical problems related to the Travelling Salesman Problem were studied in the 1800‟s by the Irish mathematician Sir William Rowan Hamilton and by the British mathematician Thomas Pennington Kirkman. Detailed discussion about the work of Hamilton & Kirkman can be seen from the book titled Graph Theory (Biggs et al. 1976).

It is believed that the general form of the TSP have been first studied by Kalr Menger in Vienna and Harvard. The problem was later promoted by Hassler, Whitney & Merrill at Princeton. A detailed description about the connection between Menger & Whitney, and the development of the TSP can be found in (Schrijver, 1960). The deterministic TSP has been studied and many algorithms has been developed[11, 13, 14, 15, 16] by many researchers.

In 2001, Applegate, Bixby, Chvátal, and Cook found the optimal tour of 15,112 cities in Germany. Later in 2004, TSP of visiting all 24,978 cities in Sweden was solved; a tour of length of approximately 72,500 kilometers was found and it was proven that no shorter tour exists. This is currently the largest solved TSP. Table 2 summarizes the milestones of solving Travelling Salesman Problem.

Table 1

Year	Research Team	Size of Instance
1954	G. Dantzig, R. Fulkerson, and S. Johnson	49 cities
1971	M. Held and R.M. Karp	64 cities
1975	P.M. Camerini, L. Fratta, and F. Maffioli	67 cities
1977	M. Grötschel	120 cities
1980	H. Crowder and M.W. Padberg	318 cities
1987	M. Padberg and G. Rinaldi	532 cities
1987	M. Grötschel and O. Holland	666 cities
1987	M. Padberg and G. Rinaldi	2,392 cities
1994	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	7,397 cities
1998	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	13,509 cities
2001	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	15,112 cities
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, and K. Helsgaun	24,978 cities

Year	Research Team	Size of Instance
1954	G. Dantzig, R. Fulkerson, and S. Johnson	49 cities
1971	M. Held and R.M. Karp	64 cities
1975	P.M. Camerini, L. Fratta, and F. Maffioli	67 cities
1977	M. Grötschel	120 cities
1980	H. Crowder and M.W. Padberg	318 cities
1987	M. Padberg and G. Rinaldi	532 cities
1987	M. Grötschel and O. Holland	666 cities
1987	M. Padberg and G. Rinaldi	2,392 cities
1994	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	7,397 cities
1998	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	13,509 cities
2001	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	15,112 cities
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, and K. Helsgaun	24,978 cities

Sources: Cook, William. "History of the TSP." *The Travelling Salesman Problem*. Oct 2009. Georgia Tech, 22 Jan 2010. <http://www.tsp.gatech.edu/index.html>. Maredia A. Described the history and implementation of TSP to solve the problem[12]

THE STRUCTURE OF TSP

Arcs, Edges and Tours

Solving a TSP amounts to finding a minimal cost Hamiltonian cycle. Recall, even though TSP is easy to understand it is very difficult to solve. Suppose we have a complete graph with nodes representing cities, and arcs representing distances between the given cities.

In this chapter Nodes represent the cities & Arcs represent distance between two cities. A complete graph with n nodes has (n-1)1/2 distinct tours or Hamiltonian cycle. As the number of nodes increases, the number of tours also increases.

Table 2: For Nodes, Arcs, Tours for (Complete Graph)

S. No.	Nodes (n)	Edges(Arcs) {(n-1)n}/2	Tours (n-1)!/2
1	1	0	0
2	2	1	1/2
3	3	3	1
4	4	6	3
5	5	10	12
6	6	15	60
7	7	21	360
8	8	28	2520
9	9	36	20160
10	10	45	181440

The number of tours increases as nodes increase

Solution Methods of TSP

A salesperson needs to travel from a city to all the other cities exactly once to sell his products and return back to the city he started from. He wants to do this while covering the minimum total distance. This is where solving the TSP comes in. Some solution methods of TSP include:

Brute – force method, Approximations - Nearest Neighbour (Greedy), Greedy Approach, Branch and bound method.

BRUTE-FORCE METHOD FOR CALCULATE SHORTEST DISTANCE

When one thinks of solving TSP, the first method that might come to mind is a brute-force method. The brute-force method is to simply generate all possible tours and compute their distances. The shortest tour is thus the optimal tour. To solve TSP using Brute-force method we can use the following steps:

Step 1: Calculate the total number of tours by (n-1)! /2 where n represent city.

Step 2: Draw and list all the possible tours.

Step 3: Calculate the distance of each tour.

Step 4: Choose the shortest tour; this is the optimal solution.

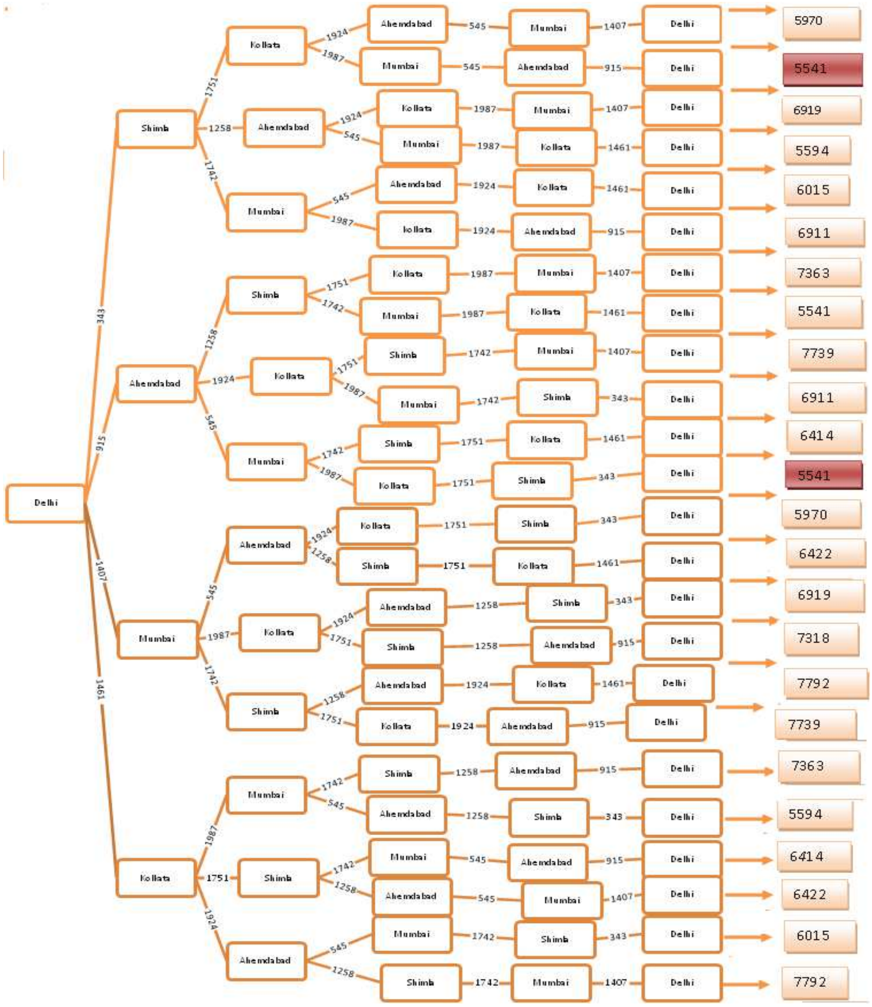
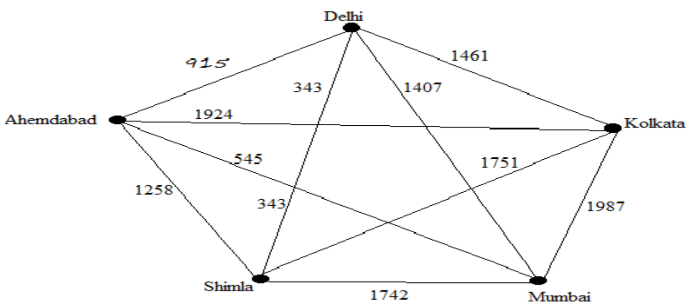


Figure 1

Complete Graph for 5 Cities

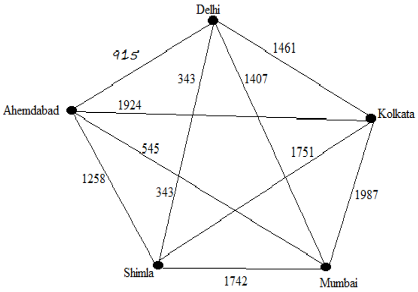


A 5 city TSP

Figure 2

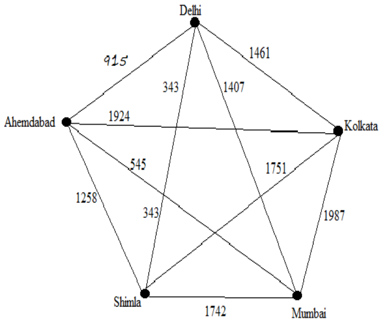
Step 1: Numbers of tours= (n-1)! /2. Here n=5 so

Numbers of tours = (5-1)! /2=4! /2=24/2=12Step: 2 and 3

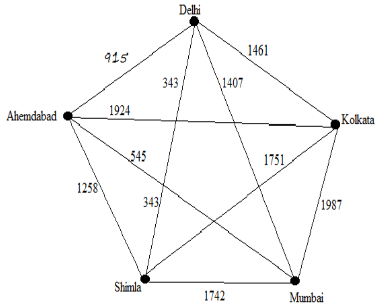


Tour 1: Delhi-Kolkata-Mumbai-Shimla-Ahemdabad-Delhi

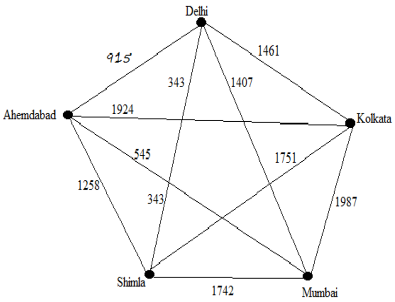
Distance of tour 1: 1461+1987+1742+1258+915=7363



Tour 2:Delhi-Kolkata-Shimla-Mumbai-Ahemdabad-Delhi
Distance of tour 2: 1461+1751+1742+545+915=6414

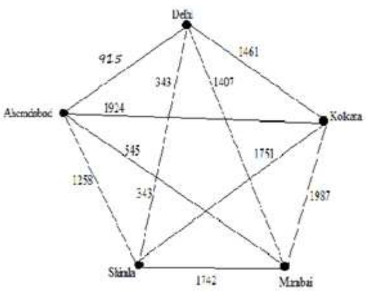


Tour 3: Delhi-Kolkata-Ahemdabad-Shimla-Mumbai-Delhi
Distance of Tour:1461+1924+1258+1742+1407=7792



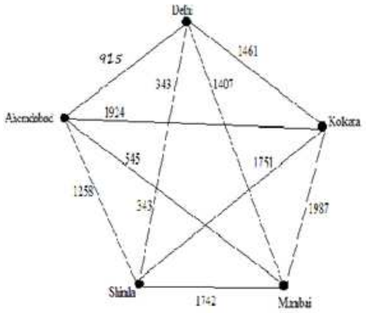
Tour 4:Delhi-Mumbai-Kolkata-Ahemdabad-shimla-Delhi

Distance of Tour:1407+1987+1924+1258+343=6919

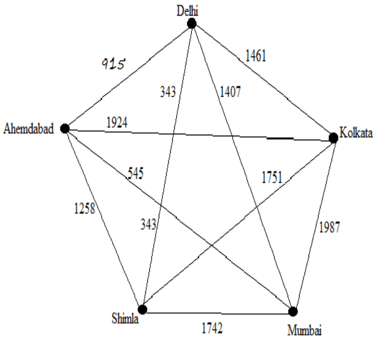


Tour 5: Delhi-Mumbai-Ahemdabad-Shimla-Kolkata-Delhi

Distance of Tour 5:1407+545+1258+1751+1461=6422

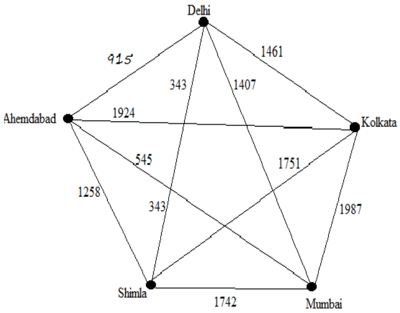


Tour 6:Delhi-Mumbai-Shimla-Kolkata-Ahemdabad-Delhi
Distance of Tour: 1407+1742+1751+1924+915=7739



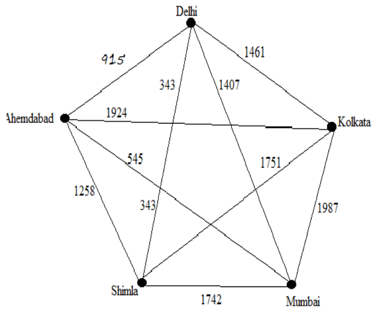
Tour 7:Delhi-Kolkata-Ahemdabad-Mumbai-Shimla-Delhi

Distance of Tour 7:1461+1924+545+1742+343=6015



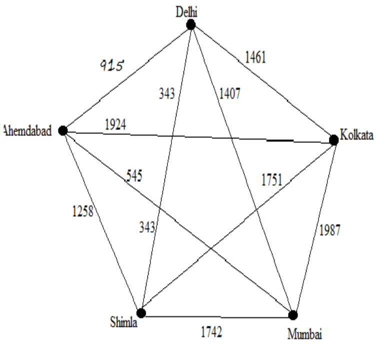
Tour 8:Delhi-Ahemdabad-Shimla-Kolkata-Mumbai-Delhi

Distance of Tour: 915+1258+1751+1987+1407=7318



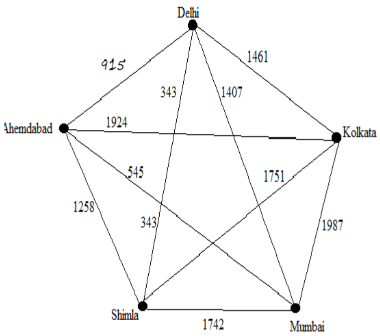
Tour 9:Delhi-Shimla-Kolkata-Ahemdabad-Mumbai-Delhi

Distance of Tour: 343+1751+1924+545+1407=5970



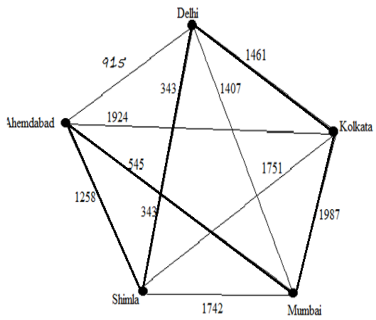
Tour 10:Delhi-Shimla-Mumbai-Kolkata-Ahemdabad-Delhi

Distance of Tour 10:343+1742+1987+1924+915=6911



Tour11:Delhi- Ahemdabad-Mumbai-Kolkata-Shimla-Delhi

Distance of Tour11:915+545+1987+1751+343=5541



Tour12 Delhi- shimla-Ahemdabad-Mumbai-Kolkata-Delhi

Distance of Tour 12:343+1258+545+1987+1461=5594

Figure 3

From analyzing all of the twelve tours we see that it will be best to take tour 11: Delhi-Ahemdabad-Mumbai-Kolkata-Shimla-Delhi. With the minimum total distance of **5541.**

Brute-Force Method for Cost of Travel

Now we calculate the fare of tour. For this we apply the same process. The chart of fare for each city shown below:

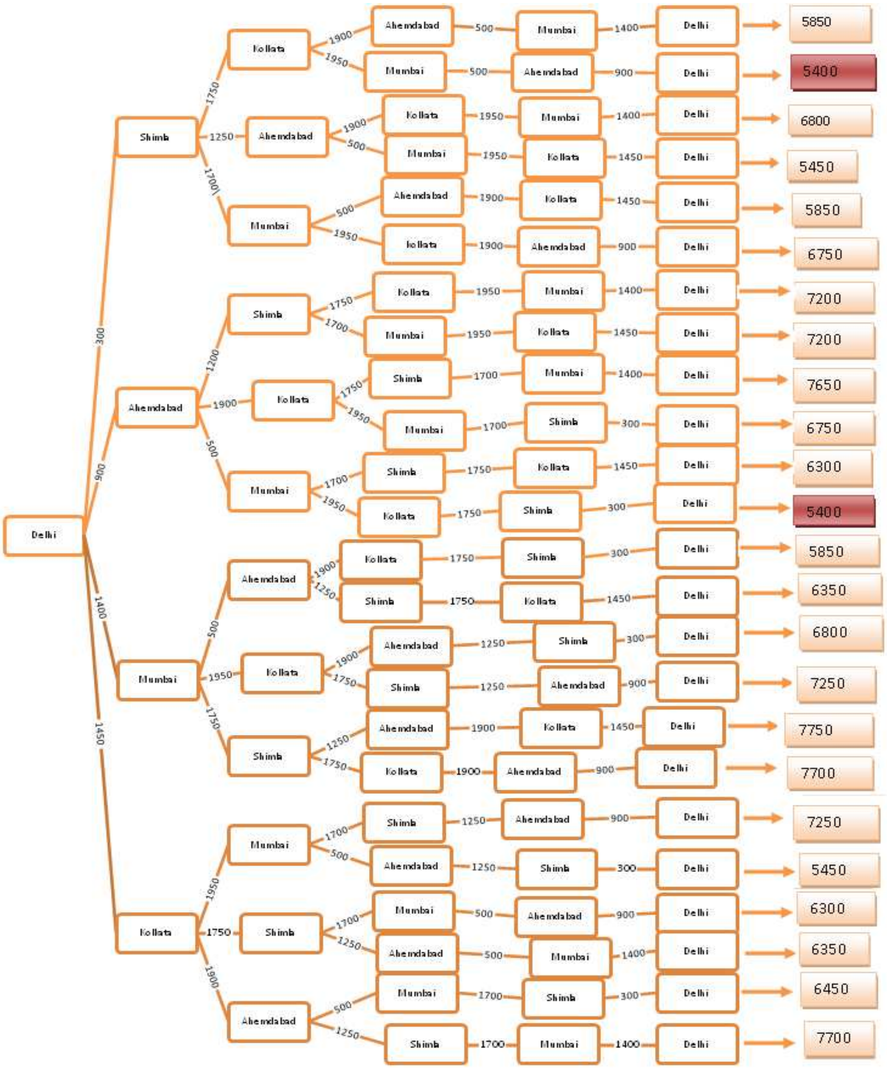
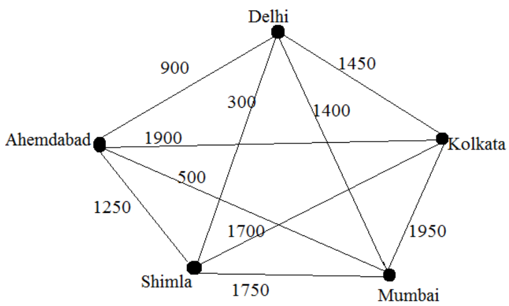


Figure 4

According to Chart, the Complete Graph for 5 City Fare Rate is as Follow

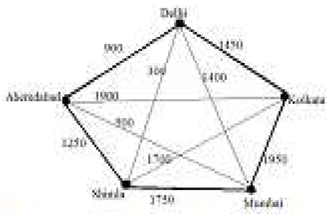


Complete graph for 5 city Fare Rate

Figure 5

Now We Calculate the Minimum Fare for Require Tour

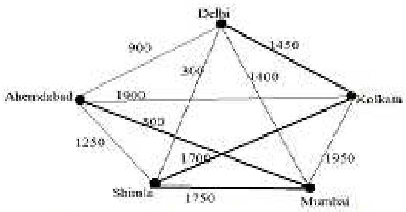
Cost of Tour 1



Tour 1: Delhi-Kolkata-Mumbai-Shimla-Ahemdabad-Delhi
cost of tour 1: 1450+1950+1750+1250+900=7300

Figure 6

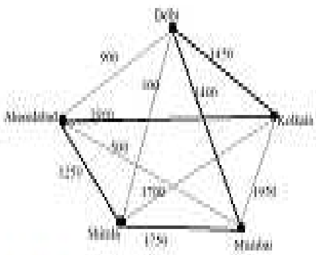
Cost of Tour 2



Tour 2: Delhi-Kolkata-Shimla-Mumbai-Ahemdabad-Delhi
Cost of Tour=1450+1700+1750+500+900=6300

Figure 7

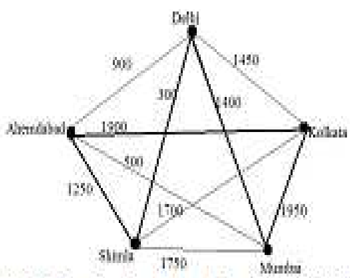
Cost of Tour 3



Tour 3: Delhi-Kolkata-Ahemdabad-Shimla-Mumbai-Delhi
Cost of Tour 3=1450+1900+1250+1750+1400=7700

Figure 8

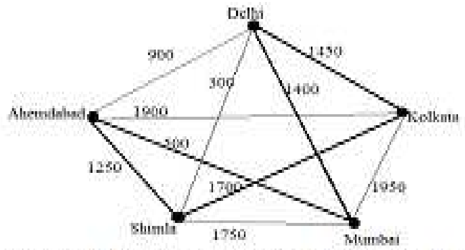
Cost of Tour 4



Tour 4: Delhi-Mumbai-kolkata-Ahemdabad-Shimla-Delhi
Cost of Tour 4=1400+1950+1900+1250+300=6800

Figure 9

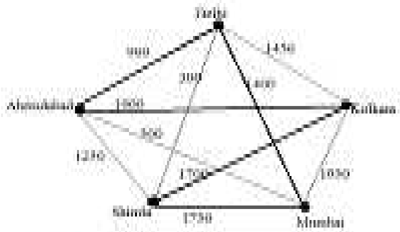
Cost of Tour 5



Tour 5: Delhi-Mumbai-Ahemdabad-shimla-Kolkata-Delhi
Cost of Tour=1400+500+1250+1700+1450=6250

Figure 10

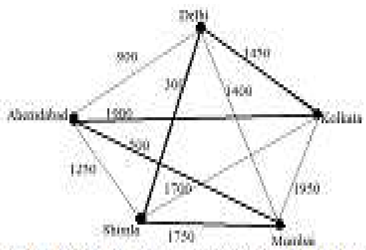
Cost of Tour 6



Tour6: Delhi-Mumbai-Shimla-Kokata-Ahemdabad-Delhi
Cost of Tour=1400+1750+1700+1900+900=7650

Figure 11

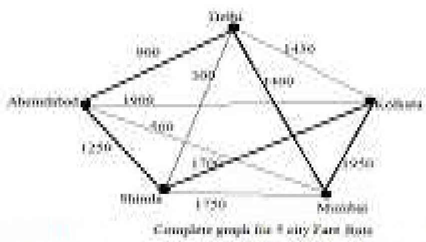
Cost of Tour 7



Tour 7: Delhi-Kolkata-Ahemdabad-Mumbai-Shimla-Delhi
Cost of Tour=1450+1900+500+1750+300=5900

Figure 12

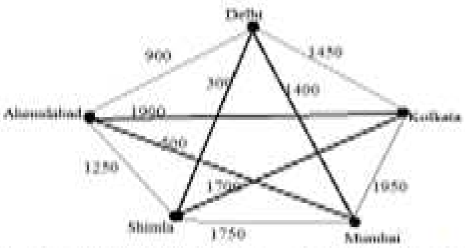
Cost of Tour 8



Tour 8: Delhi-Ahemdabad-Shimla-Kolkata-Mumbai-Delhi
Cost of Tour=900+1250+1700+1950+1400=7200

Figure 13

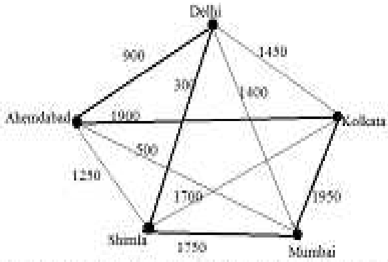
Cost of Tour 9



Tour 9: Delhi-Shimla-Kolkata-Ahemdabad-Mumbai-Delhi
Cost of Tour=300+1700+1900+500+1400=5800

Figure 14

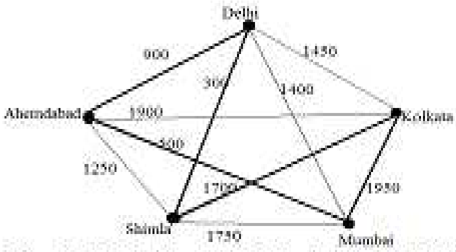
Cost of Tour10



Tour 10: Delhi-Shimla-Mumbai-Kolkata-Ahemdabad-Delhi
Cost of Tour=300+1750+1950+1900+900=6800

Figure 15

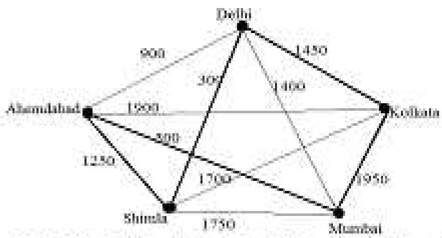
Cost of Tour 11



Tour 11: Delhi- Ahmadabad -Mumbai- Sheila -Delhi
Cost of Tour=900+500+1950+1700+300=5350

Figure 16

Cost of Tour 12



Tour 12: Delhi- Shimla -Ahemdabad- Mumbai –Kolkata-Delhi
Cost of Tour=300+1250+500+1950+1450=5400

Figure 17

By the above graph of tour we see that the shortest path with minimum fare is Tour 11: Delhi- Ahemdabad -Mumbai- Shimla –Delhi. Cost of Tour=900+500+1950+1700+300=5350

Brute –force method always works if given enough time and care. Because of its nature, it is convent for relatively small number of nodes.

Recall from table as given below we see that as the number of nodes increases the number of tours increases factorially; therefore, using Brute force method to solve TSP with a large number of nodes can be frustrating and can even take years of centuries to solve.

Table 3: Shows How Rapidly the Number of Tour Increases as Number of City Increases

S. No.	Number of City (N)	Number of Arcs = N(N-1)/2	Numbers of Tours= (N-1)1/2
1	6	15	60
2	7	21	360
3	8	28	2520
4	9	36	20160
5	10	45	181440
6	11	55	1814400
7	12	66	19958400
8	13	78	239500800
9	14	91	3113510400
10	15	105	43589145600

The number of Tours increases as n increases

So we cannot solve TSP for a large number of nodes. We use nearest neighbour algorithm to solve TSP.

Nearest Neighbour

The nearest neighbour heuristic, is a simple approach for solving the travelling salesman problem. To solve TSP with a Nearest Neighbour Heuristic we look at all the arc coming out of the city that have not been visited and the next closet city.

To solve TSP with a Nearest Neighbour heuristic we look at all the arcs coming out of the city (node) that have not been visited and choose the next closest city, then return to the starting city when all the other cities are visited. To solve TSP using Nearest Neighbour Heuristic we can use the following steps:

Step 1: Pick any starting node.

Step 2: Look at all the arcs coming out of the starting node that have not been visited and choose the next closest node.

Step 3: Repeat the process until all the nodes have been visited at least once.

Step 4: Check and see if all nodes are visited. If so return to the starting point which gives us a tour.

Step 5: Draw and write down the tour, and calculate the distance of the tour.

Calculate TSP Using Nearest Neighbour Heuristic

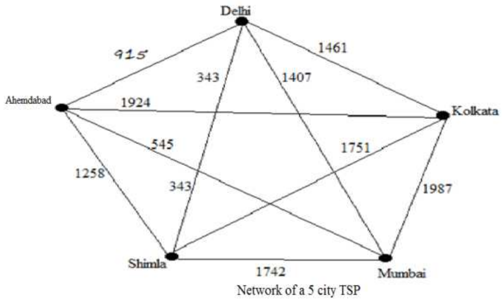


Figure 18

Step 1: We start with city Delhi.

Step 2: There are four arcs coming out of city Delhi, first one is arc Delhi-Kolkata with distance of 1461, second one is arc Delhi-Mumbai with distance of 1407, and the third one is arc Delhi-Shimla with distance of 343, and the fourth one is arc Delhi-Ahemdabad with distance of 915. The next closest city is Shimla with the distance of 343.

Step 3: From city Shimla, we have four arcs coming out. We cannot go back to Delhi. Since we have already visited that city. We can either go to city Ahemdabad, city Mumbai or city Kolkata. We pick city Ahemdabad since the next closest city from city Shimla is Ahemdabad with the distance of 1258. Now from Ahemdabad we cannot go back to city Delhi and city Shimla, since we have already visited those cities. Therefore The closest city from city Ahemdabad is Mumbai with the distance of 545, and from Mumbai the next closest city is Kolkata with the distance of 1987.

Step 4: Since we have visited all the cities, now we can finally return from city Kolkata to city Delhi which has the distance of 1461.

Step 5:

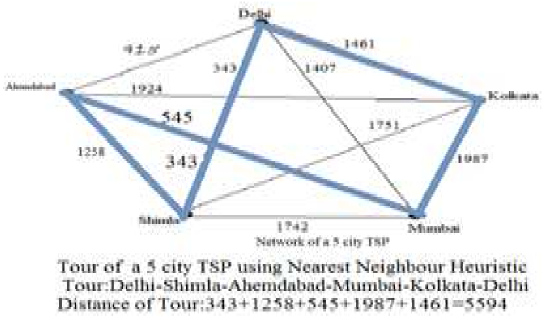


Figure 19

Nearest Neighbour Heuristic for Cost of Travel

We first choose the city Delhi-Shimla with rate of fare is 300. Second we choose Shimla-Ahemdabad with fare 1250. Then we choose Ahemdabad-Mumbai with fare 500. And then we choose Mumbai-Kolkata with Fare 1950. Finally we choose Kolkata-Delhi with fare 1450. If we add the entire rate of fare of tour then we have 5450 rupees.

By comparing the result of example brute force method and nearest neighbour heuristic. We see that the optimal solution for Brute-force method is not same. The optimal solution of Nearest Neighbor heuristic is 5594. While the optimal solution of brute force method is 5541. Here Nearest Neighbour not worked, but it will not always be the case. Therefore, it follows that Nearest Neighbour Heuristic works but it does not always give you the best solution as Brute-force method.

Greedy Approach

The Greedy Approach can be the first method which can be used to solve TSP. To solve TSP using Greedy Approach, we look at all the arcs coming out of the city (node) and choose the n cheapest arcs. If those n cheapest arcs forms a Hamiltonian cycle than we have an optimal solution. To solve TSP using Greedy Approach we can use following steps:

Step 1: Look at all the arcs with minimum distance.

Step 2: Choose the *n* cheapest arcs

Step 3: List the distance of arcs starting from the minimum distance to maximum distance.

Step 4: Draw and check if it forms a Hamiltonian cycle.

Step 5: If step 4 forms a Hamiltonian cycle than we have an optimal solution; write down the tour of the optimal solution and calculate their distance.

A 5 City TSP Network

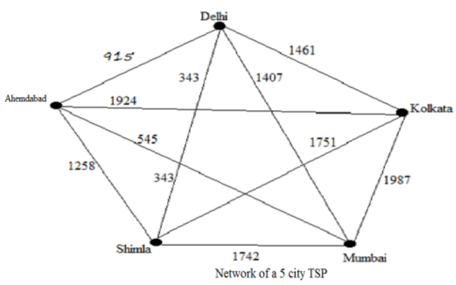


Figure 20

Step 1: the arcs with minimum distances are arc Delhi-Shimla, arc Ahemdabad-Mumbai, arc Delhi-Ahemdabad, arc Ahemdabad-Shimla, arc Delhi-Kolkata.

Step 2: we choose *n* cheapest arcs and they are DB, BA, AC, and CB.

Step 3: arc DB = 1, the next smallest distance is BA = 3, arc AC = 4 is the next smallest distance, and the fourth smallest distance is arc CB = 5. We stop at arc CB because we already picked *n* cheapest arcs.

Step 4 and 5:

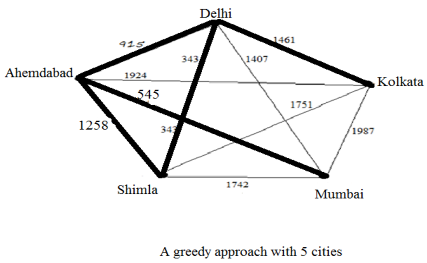


Figure 21

We do not have a Hamiltonian cycle, therefore, we do not have a tour. We conclude that this method is a good approach, but it does not give us the optimal solution. Therefore the greedy approach is not a good approximation.

CONCLUSIONS

Graph theory is very useful to solve TSP. TSP is solved by many another method. Here we used Brute Force Method, Nearest Neighbour Algorithm, Greedy Algorithm to solve the problem. The result comes from these three methods indicate that Brute force method is most useful to solve the TSP for small plan of travel.

REFERENCES

1. Adamchik, Victor. "Graph Theory". 2 April 2010 <http://www.cs.cmu.edu/~adamchik/21- 127/lectures/graphs_4_print.pdf>.
2. Arora S, Polynomial Time Approximation Schemes for Euclidian Traveling Salesman and Other Geometric Problems, *Journal of the ACM.*, vol. 45, No. 5, pp.753-782, 1998.
3. Bentley J J, Fast Algorithms for Geometric Traveling Salesman Problem, *ORSA J. Com-put.*, vol. 4, No. 4, pp.387-411, 1992.
4. Bertsekas, Dimitri. Network Optimization: Continuous and Discrete Models. Belmont: Dimitri P. Bertsekas, 1998.
5. Boffey, T. Graph Theory in Operations Research. 1st ed. London and Basingstoke: The Macmillan Press Ltd, 1982. pp. 148-185. 25 November 2009.
6. Bondy J A, Murty U S R, *Graph Theory*, Springer, 2008.
7. Bronson, Richard, and Naadimuthu Govindasami. Operations Research. New York: McGraw-Hill, 1982
8. Calgor, H. "The Brute Force Algorithm". 24 January 2010 <http://www.ctl.ua.edu/math103/hamilton/HCalgor.htm#THE BRUTE FORCE ALGORITHM>.
9. Cook, William. "History of the TSP." *The Travelling Salesman Problem*. Oct 2009. Georgia Tech, 22 Jan 2010. <http://www.tsp.gatech.edu/index.html>.
10. Dantzig G, Fulkerson R, Johnson S, Solution of A Large-Scale Traveling-Salesman Prob-lem, *Operations Research*, vol. 2, pp.393-410, 1954.
11. Lin S, Computer Solutions for The Traveling Salesman Problem, *Bell Syst. Tech J.*, vol. 44, pp. 2245-2269, 1965.
12. Maredia A, History, Analysis and implementation of Travelling Salesman Problem(TSP) and Related Problems, Thesis, University of Houston-Downtown, pp. (1-40), Spring-2010.
13. Papadimitriou C H, Yannakakis M, The Traveling Salesman Problem with Distances One and Two, *Mathematics of Operations Research*, vol. 18, No. 1, pp. 1-11, 1993.
14. Platzman L K, Bartholdi J J, Space¯lling Curves and The Planar Travelling Salesman Problem, *Journal of the ACM.*, vol. 36, No. 4, pp. 719-737, 1989.

15. Reinelt G, TSPLIB-A Traveling Salesman Problem Library, *ORSA J. Comput.*, vol. 3, No. 4, pp.376-384, 1991.
16. Rosenkrantz D J, Stearns R E, Lewis P M, An Analysis of Several Heuristics for The Traveling Salesman Problem, *SIAM J. Comput.*, Vol. 6, No. 3, pp.563-581, 1977.
17. Wilson, Robin. Introduction to graph theory. 4th ed. England: Addison-Wesley, 1996. Pp.166. 19 January 2010.