APPLICATION OF BRUTE FORCE ALGORITHM ON DISTRICTS ARTERY NET

K. Perarasan [1], D.R.Kirubhakaran [2],J.Christina Suvakeen[3]

[1]Assistant Professor, Department of Mathematics, Government College of Engineering, Thanjavur,Tamilnadu, India, rinomathz502@yahoo.com
[2] Assistant Professor, Department of Mathematics, A. V. V. M. Sri Pushpam College, Poondi, Thanjavur, Tamilnadu, india, jagkans@yahoo.com
[3]Assistant Professor, Department of Mathematics, Parisutham Institute of Technology, Thanjavur, Tamilnadu, india, tina.suvakeen@gmail.com

ABSTRACT

This paper delivers a manifestation of spanning algorithms by creating a "travel plan" between 5 places Selam, Kirishnagiri, Vellore, Bangalore, and Mysore. Our objective is to find the path with minimum cost and having shortest distance. It is found that the next-door neighbour algorithm (NDNA) does not give us an optimal path all the time. The Brute force algorithm in which we find out all the possibilities lead to our solution in our constraints. We involved both the cost of travelling and distance for expressing the optimal path. By Kruskal's algorithm we find the reduced path (shortest path) when we do not have any such constraints.

Keywords: Nearest-door neighbour algorithm, Brute force algorithm, Kruskal's algorithm.

1. INTRODUCTION

The basic aspire of Graph theory is to solve problems related to a network designed by interconnecting a set of points by lines. Its most significant application is travel planning. A graph may direct or undirected. Undirected graph denotes to a network of points where there is no distinction among two vertices or edges. If a node is joined by more than one edge then it is known as multi-graph. If a graph has only one edge between any two nodes, then it is simple graph. Degree of a vertex is the number of edges incident on it. Anyway, taken down the edges of a graph is a path. A path originating and terminating at the same node is known as circuit. If a circuit includes every edge in the graph, then the graph is said to be Eulerian. If a circuit visits each vertex once and ends at the origin, it is called a Hamiltonian circuit. Even though both Euler circuit and Hamiltonian cycle may seem same, every node is visited in an Eulerian circuit while in a Hamiltonian cycle each edge is traversed once

Thus, by inheriting the ideas of graph theory, we can determine which city to traverse first while planning the travel. When we use the brute force method to solve such the travel planning problem, even if it returns an optimal solution, it is largely time and space consuming particularly when information to be processed is massive. The process can't be completed manually as the number of points in large in number.

The next-door neighbour algorithm (NDNA) was one of the first algorithms used to determine a solution to the travelling salesman problem. For more efficient method for finding the optimal solution, the next-door neighbour algorithm is implemented. This method is time critical. However, it does not always return an optimal solution. For finding the shortest path in travel planning NDNA can be used. The solution is said to be optimal when the distance to be covered is shortest. However, NDNA being a heuristic algorithm cannot guarantee an optimal solution. For finding the solution using NDNA, starting from vertex, we move to the closest vertex and so on till we reach the destination vertex. Thus, by using NDNA we get a travelling plan which may or may not be optimized.

The Sorted Edges Algorithm (SEA), well known as "Travelling Salesman Problem", is a Hamiltonian Circuit as well as heuristic. The eventual objective of the travelling salesman problem is to find the least cost for traversing. In SEA, the edges are sorted according to their weight. First, the smallest edge is selected, then the next smallest and so on. If edges make a circuit, they should be rejected unless they form a Hamiltonian circuit.

In 1956, one standard algorithm has been developed by Kruskal's. Each vertex can be visited exactly once and no circuits are formed when we implement Kruskal's algorithm. Here, a set of edges which forms a tree by including all the nodes, where the cumulative cost of edges is kept lowest. Thus, it finds the optimal path.

In section 1.3 of this paper, we find the path with minimum cost and having the shortest distance using Kruskal's algorithm and the result also discussed. We conclude this paper in section 3.

1.1 Method

Our aim is to try to optimize the path based on the distance between multiple nodes which we denote as multiple districts like *Selam, Kirishnagiri,Vellore, Bangalore and Mysore*. To trace the shortest possible path between multiple nodes, we tried that by implementing two algorithms. Time and cost are taken by manual to avoid the discrepancies for the research.

Origin point	Destination point	Cost (In Rs)	Distance (In Km)
Selam	**Kirishnagri**	**205**	**115**
Selam	**Vellore**	**392**	**234**
Selam	**Bangalore**	**450**	**238**
Selam	**Mysore**	**700**	**256**
Kirishnagri	**Vellore**	**550**	**109**
Kirishnagri	**Bangalore**	**420**	**94**
Kirishnagri	**Mysore**	**1602**	**227**
Vellore	**Bangalore**	**624**	**116**
Vellore	**Mysore**	**924**	**464**

Bangalore	**Mysore**	**300**	**147**

The above scenario can be depicted as:

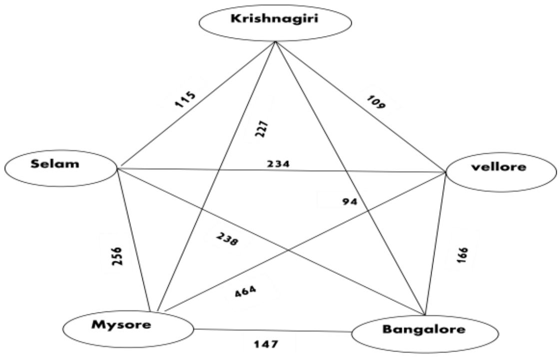

Fig.1.1 Graph based on distance between cities

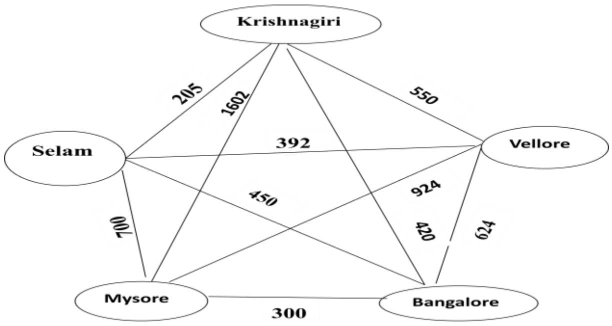

Fig 1.2 Graph based on cost between cities

1.2 Algorithm 1

Considering Selam as starting point of travelling we are going to find out the minimized shortest path based on distance as well as with cost by brute force method. It leads to finding all the possibilities of paths and then finally finding the least value path.

So comparing the cost we obtained using this brute force method we find that Selam-Vellore-Krishnakiri-Bangalore-Mysore gives the shortest path based on distance and Selam-Vellore-Krishnakiri-Bangalore-Mysore is an economical path on basis of cost.

1.3 Algorithm 2

Apply Kruskal's algorithm to find the shortest spanning tree for the given graph.
We perform following steps to trace the path:

1. Listing the edges of the graph from upper to the lowest in uphill or downhill order of its weight.
2. Selection of edge from that order in such a way that no circuit is formed.
3. If an edge forms a circuit with previously selected ones then we have to neglect that edge and look for next edge.
4. We have to draw the spanning tree of graph covering all the vertices.

So we obtained following two spanning trees based on their edge distances and cost of travelling.

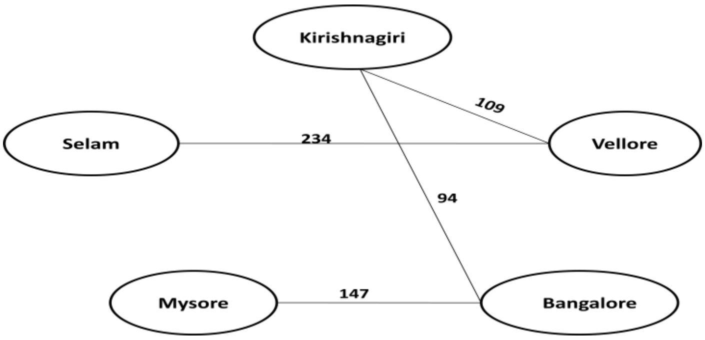

Fig 1.3 Shortest path based on distance.

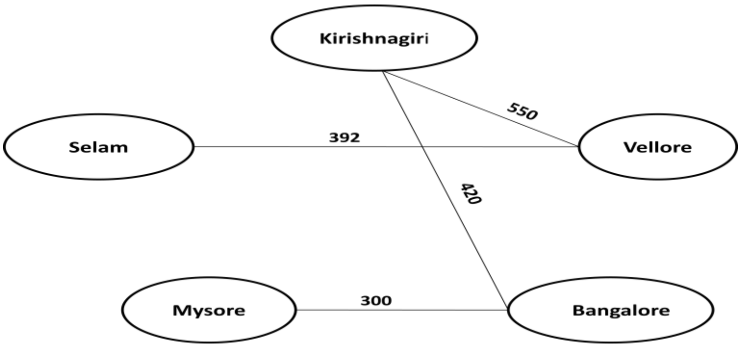

Fig 1.4 Cost-effective path based on cost.

1.4 Algorithm 3

Now we will be representing the next-door path algorithm for travelling and finding the path. So we start from Selam and will try to cover all the Places and come back to Selam. Now considering the case of distance graph, we start our simulation from Selam. We find that Vellore is the nearest city closest to it so we move to Vellore Similarly continuing in this way; the path obtained will be: Selam Kirishnagiri Vellore Bangalore Mysore Selam

IJRISE International Journal of Research In Science & Engineering e-ISSN: 2394-8299
Volume: 3 Issue:2 March-April 2017 p-ISSN: 2394-8280

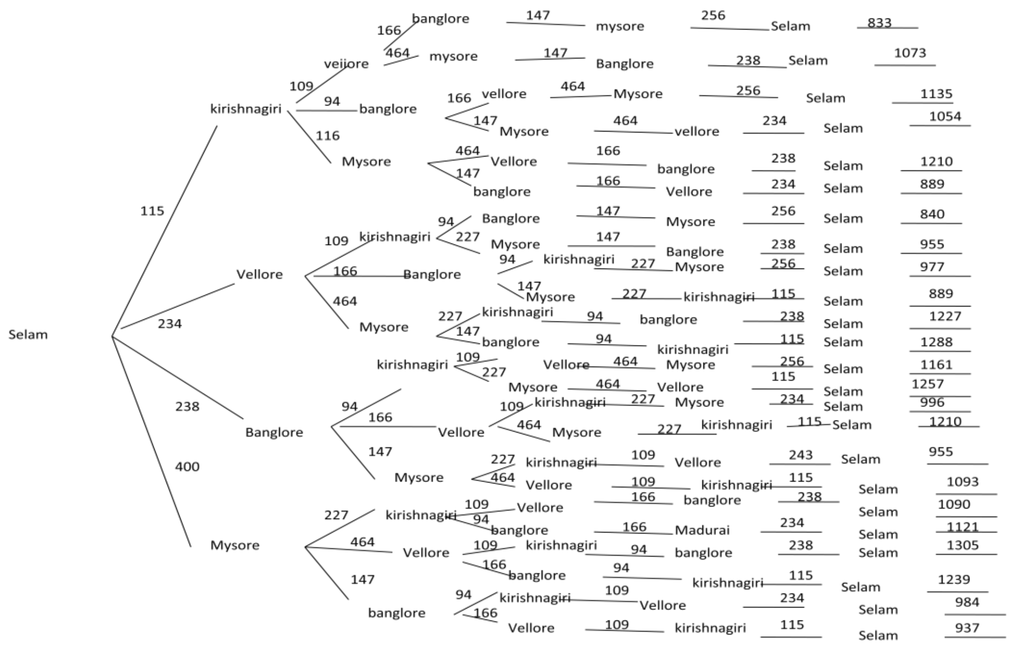

Fig 1.5 Brute force on basis of Distance

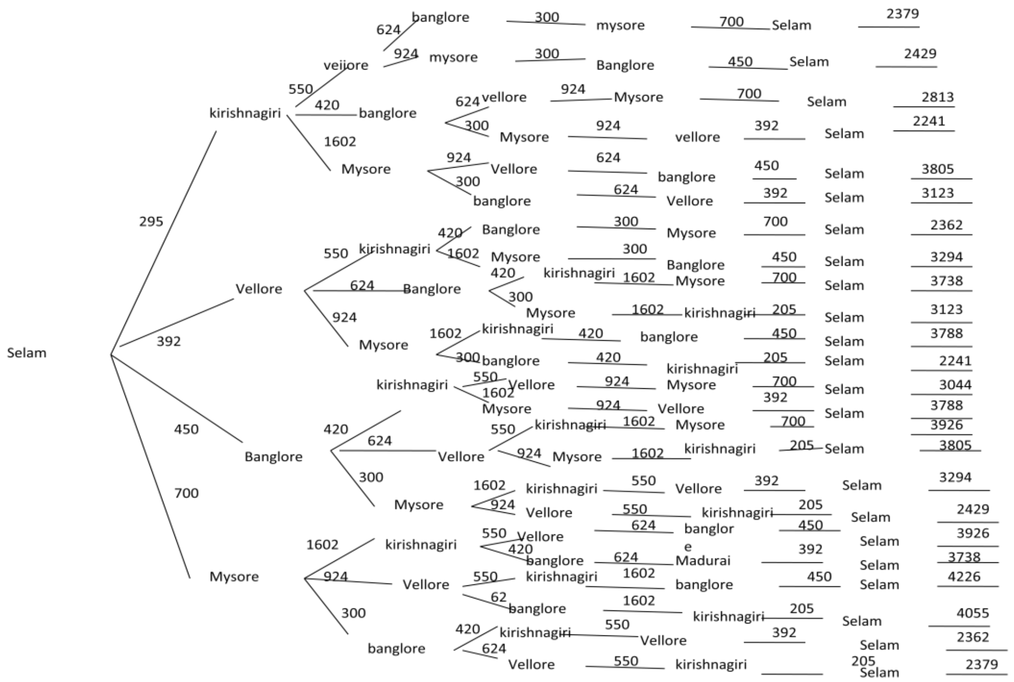

Fig 1.6 Brute force on basis of Cost

The distance, weight of travelling path obtained is 833. To calculate the cost of travelling using this next-door node algorithm. So starting from Selam, we move to Vellore since the travel cost is lowest for it. Continuing in this way we obtain following path:- Selam-Kirishnagiri- Vellore- Bangalore- Mysore- Selam the travel cost obtained is 2379.

2. RESULTS

By using the concepts of graph theory through various graphs obtained by applying multiple algorithms to find out the optimal path, we found that Selam Kirishnagiri Vellore Bangalore Mysore Selam is most optimized path with path distance 833 km and cost of travelling Rs. 2379. We took Omni Bus fare between the cities for the simulation. So according this we can easily plan our travelling program in most optimized way saving our money as well as time.

3. CONCLUSION

This paper gave us an idea about how we can engagement concepts of Graph theory in our day to day life. We find out the most optimized path between five cities in terms of time and money. It denotes the power of use of Graph theory as it can be obliging to our daily necessities.

We conclude that NDNA (next-door neighbour algorithm) does not give us optimal path all the time as we got different values of travel cost and path using that algorithm. Brute force algorithm in which we tried to find out all the possibilities lead to our solution in our constraints. We involved both the cost of travelling and distance for expressing the optimal path. We can use the concepts of Kruskal's algorithm to find the minimized path (shortest path) when we do not have any such constraints.

REFERENCES

[1] Rodriguez, B (1998-2009). Graph Theory. Retrieved July 15, 2009, from Applications of Graph Theory.

[2] West, D (2005). Graph Theory. Retrieved July 15, 2009, from Douglas B. West's