Application Of Graph Theory To Find Optimal Path And Minimized Cost For The Transportation Problem

[1]Priyadharshini.T,[2]Manikandan.K.M

[1]PG Student, [2]Associate Professor, Department of Mathematics, Dr.SNS Rajalakshmi College of Arts and Science, Coimbatore,Tamil Nadu, India

ABSTRACT *- Graph theory is used to model many types of relations and process in real world network systems such as internet, power grids, telephone, transportation, affiliation networks, food webs, electricity, water and so on. Many practical problems can be represented by graphs. In this paper we study about finding minimum spanning tree and the shortest distance between two place using algorithms. We have developed a network model to find the minimum distance and the minimum cost of transporting a product from place A to place B through a truck.*

Keywords: *Spanning tree, vertices, edges, algorithms, place, distance, cost.*

I. INTRODUCTION TO GRAPH THEORY

A graph G is defined as a set of vertices called nodes 'V' which are connected by edges called links 'e'. Thus $G = (V, e)$. A vertex (node) v is an intersection point of a graph. It denotes a location such as city, a road intersection or a transport terminal(stations, harbours and airports).A edge 'e' is a link between two vertices (nodes). Transport geography can be defined by a graph. Most networks, namely road, transit and rail networks are defined more by their link(edge) than by nodes(vertex). In this paper graphs are finite, simple, connected and undirected graph.

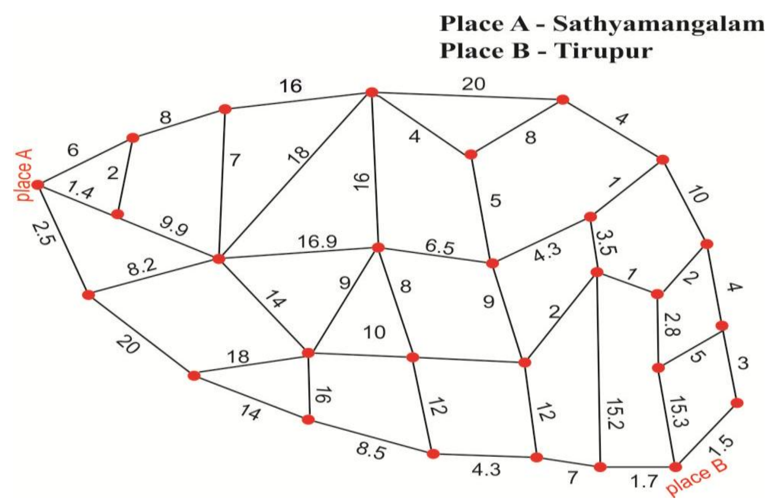

Figure 1: Connected Graph

There are different ways to reach from place A to place B. Here we find the shortest distance between A to B with minimum costs.

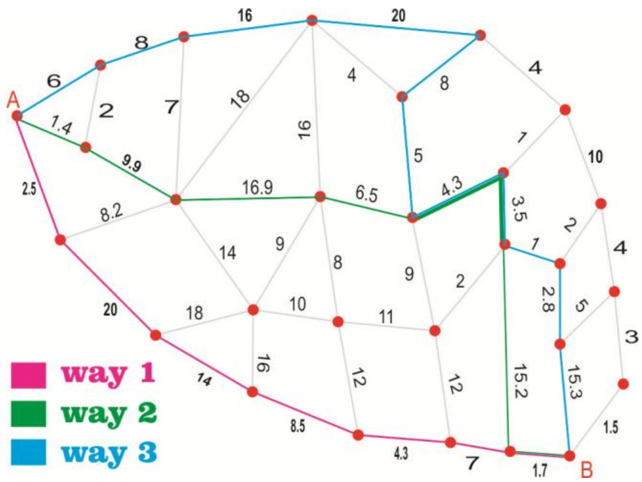

Figure 2: Different ways to reach place A to place B

Here we can find three different ways to reach the places from place A to place B.

II. MINIMUM SPANNING TREE [MST]

A minimum spanning tree for a weighted, connected and undirected graph is a spanning tree with weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

III. KRUSKAL'S ALGORITHM

Spanning tree to find the shortest distance between to two places from A to B.
Step 1: Remove all loops and parallel edges
Step 2: Arrange all the edges in ascending order of distance in kilometers.
Step 3: Add edges with least weight.

IV. MINIMUM SPANNING TREE [MST] USING KRUSKAL'S ALGORITHM

.

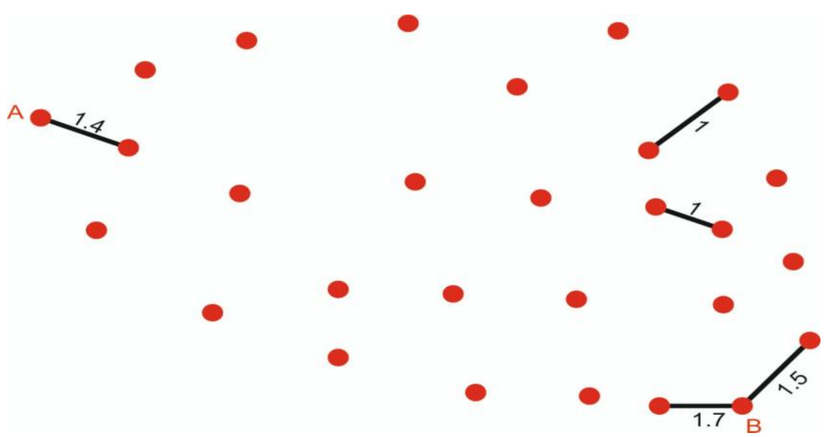

Figure 3: Minimum Spanning tree for 1<2

In this paper the minimum spanning tree for the given graph is described by several figures. First we will start to put the edges in their place A starting from the shortest distance to the highest distance to the place B. Here we taken the distance as kilometers and we find a condition that 1<2 that is we consider the distance from 1 to 1.9 for the given figure.

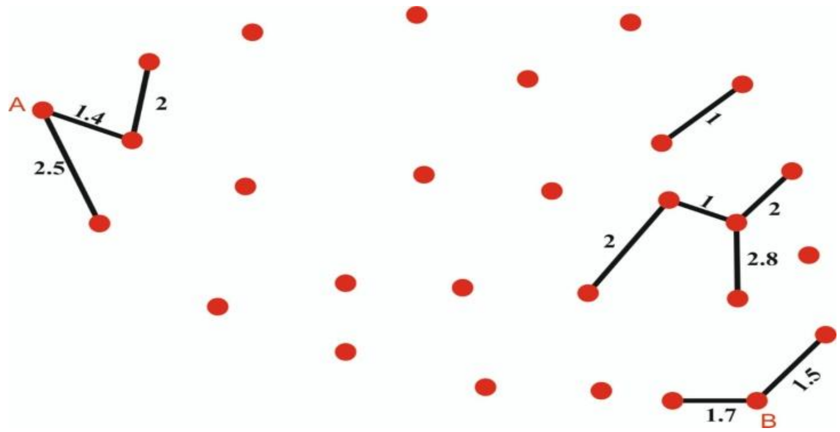

Figure 4: Minimum spanning tree for 2<3

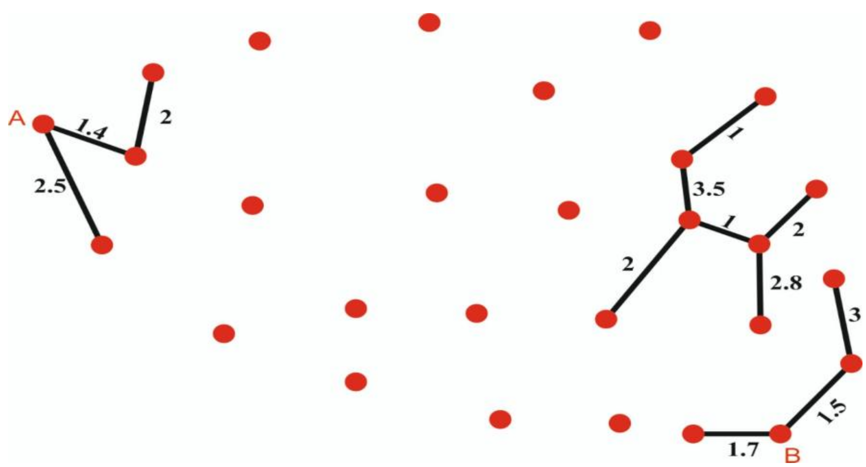

Figure 5: Minimum spanning tree for 3<4

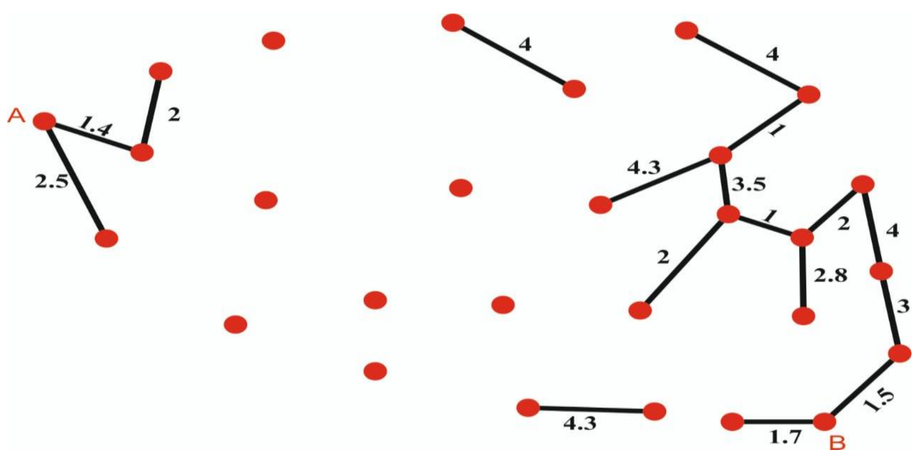

Figure 6: Minimum spanning tree for 4<5

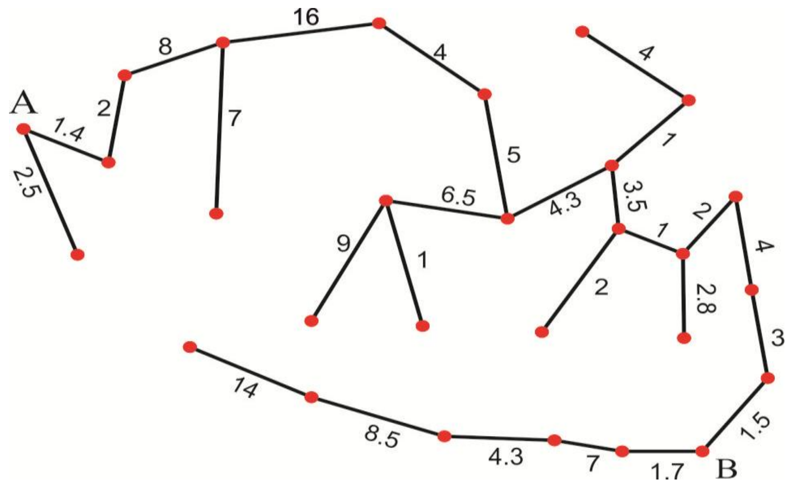

Figure 7: MST

In this above figure we have find the minimum spanning tree which it doesn't form a cycle. Since a tree is acyclic in nature.

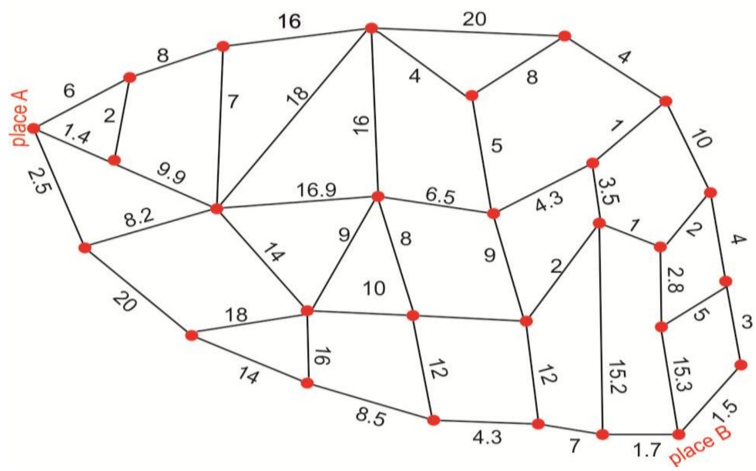

Figure 8: Minimum Spanning Tree

Applying this algorithm to all the edges, we obtain a minimum spanning tree as shown in the figure7.

V. MINIMUM COST PATH [MCP]

From the minimum spanning tree as shown in the figure, we find the minimum cost from place A to place B.

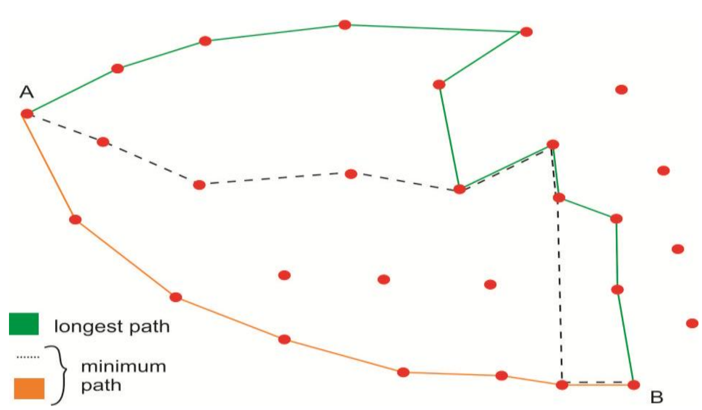

Figure 9

1) **First option -** To calculate the distance from place A to place B (orange full line), the result is:
$\delta = 2.5 + 20 + 14 + 8.5 + 4.3 + 7 + 1.7 = 58\ km$ (less distance)

2) **Second Option** (dash line)
$\delta = 1.4 + 9.9 + 16.9 + 6.5 + 4.3 + 3.5 + 15.2 + 1.7 = 59.4\ km$ (dash line)

3) **Third Option**
$\delta = 6 + 8 + 16 + 20 + 8 + 5 + 4.3 + 3.5 + 1 + 2.8 + 15.3 = 89.9 km$ (green line)

The first option represents the minimum cost path from place A to place B (Sathyamangalam to Tirupur).

VI. COMPLEXITY

We consider all the three options and we observe that third option has the longest path so we are not considering the third option. Here we compare the two paths with the MST and minimum cost path. We choose the first option because it covers minimum distance of 58km than 59.4km. The minimum cost path is denoted by orange line in figure 9.

VII. PRIM'S ALGORITHM

By using Prim's Algorithm we are able to find the shortest distance from a vertex (place A) to all other vertices to place B.

In this paper the MST for the given case is described by following figures. First we start from place A, which is chosen as permanent and assign its value 0. Assign the distance (kilometers) of the neighborhood vertices of the place A, to find the shortest path to vertex 2 which is equal to 1.4. Now vertex 2 is chosen as permanent and check the distance from vertex 2 to the neighbor vertices, to each neighbor vertex is add the length of the permanent vertex.

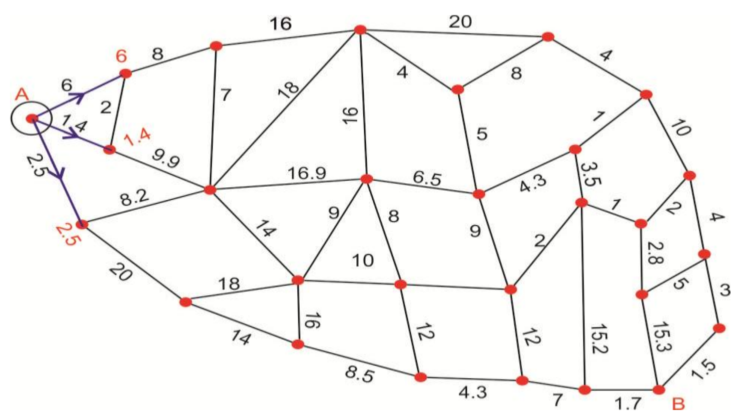

Figure 10: Minimum Path

From vertex A we start to find the values of each vertex $\Rightarrow 0 + 1.4 = 1.4\,;\ 0 + 6 = 6\,;$

$$0 + 2.5 = 2.5$$

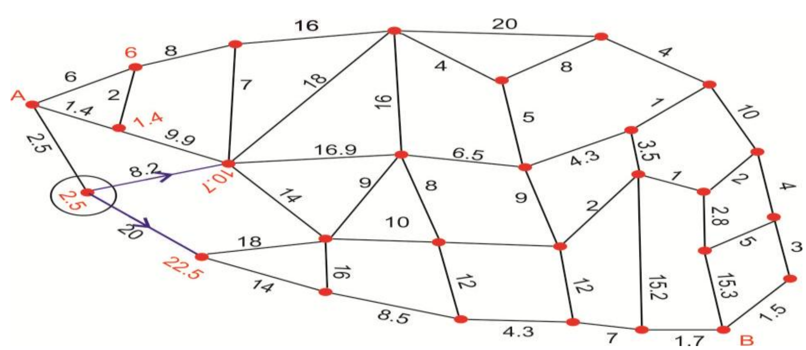

Figure 11: Minimum Path

We have chosen the minimum distance from vertex A. $1.4 + 9.9 = 11.3$ & $2.5 + 8.2 = 10.7$

$$\text{Minimum distance is chosen as a permanent vertex, since the distance } 10.7 \textit{ is shorter than } 11.3$$
$$\Rightarrow \textit{distance } 9.9 \textit{ is not considered anymore}$$

Now we have to consider the distance 9.9

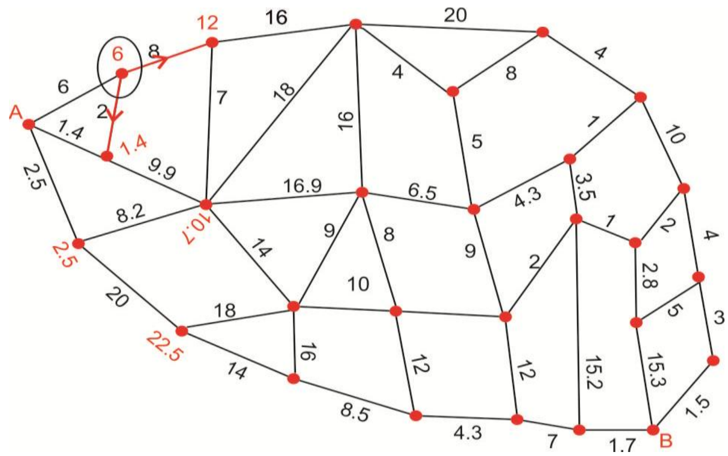

Figure 10: Minimum Path

This process is repeated for each vertex respectively.

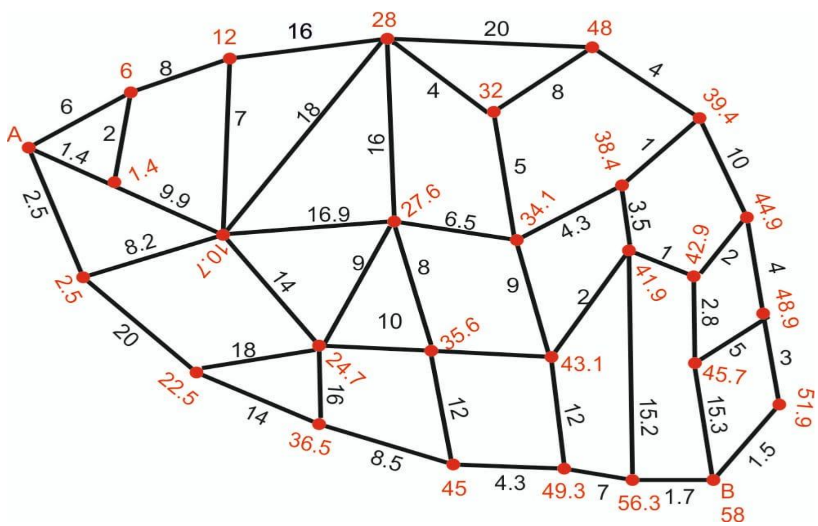

Figure 12: Minimum Path

Finally we find all the values of the vertices in minimum spanning tree with distance of 58 kilometers.

VIII. MINIMUM PATH BETWEEN VERTICES OF PLACE A TO PLACE B

Now we are going to find the shortest path from vertex A to vertex B, for every adjacent vertex v, if weight of edge u-v is less than the previous value of v, then update the value as weight of u-v.

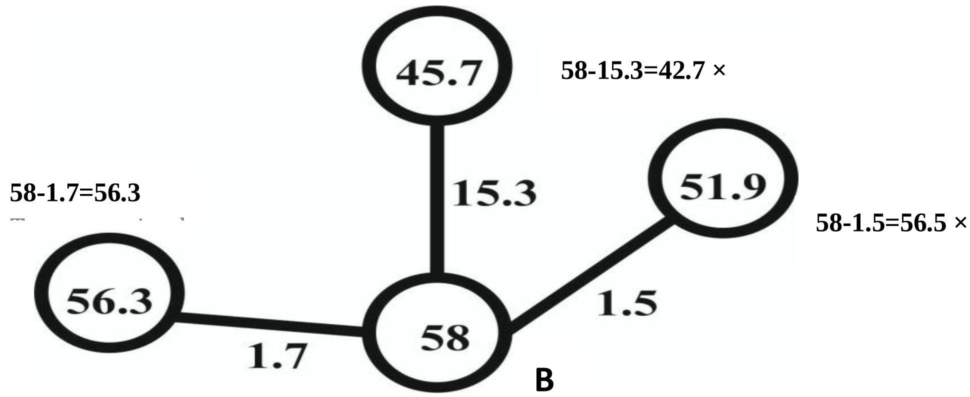

Figure 13

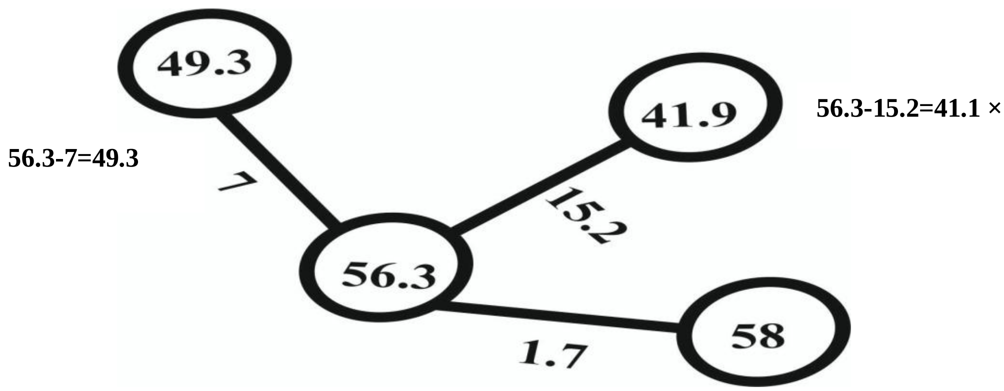

Figure 14

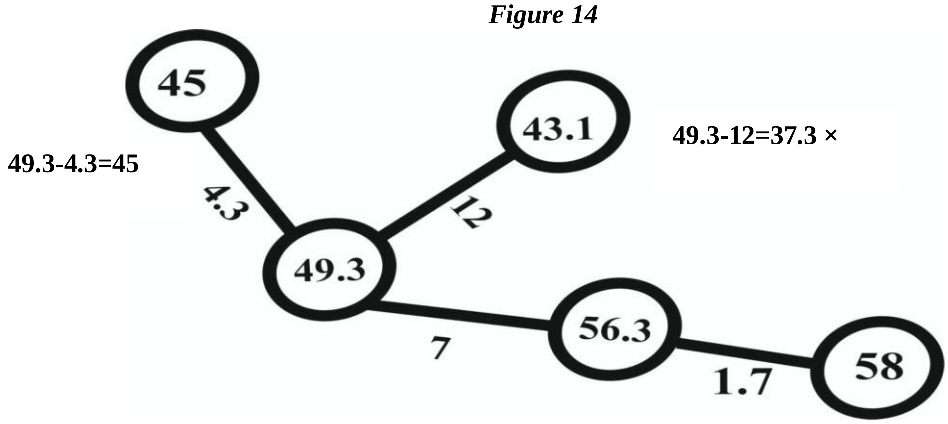

Figure 15

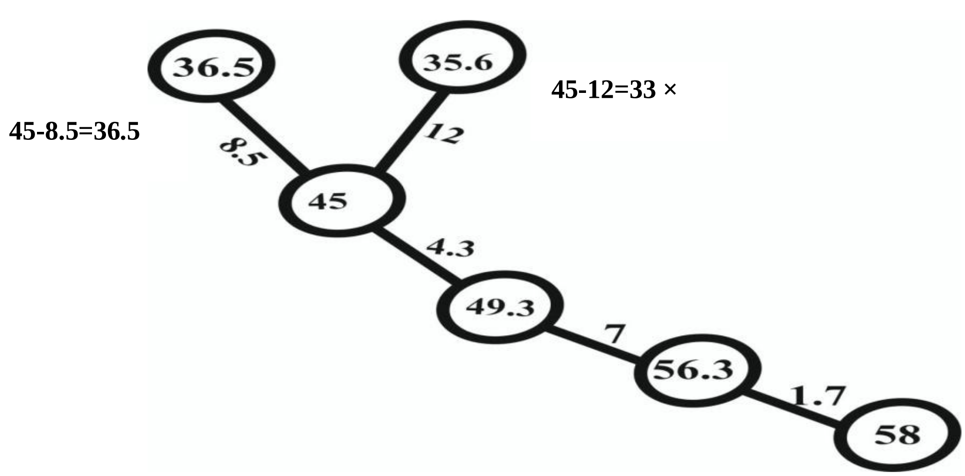

Figure 16

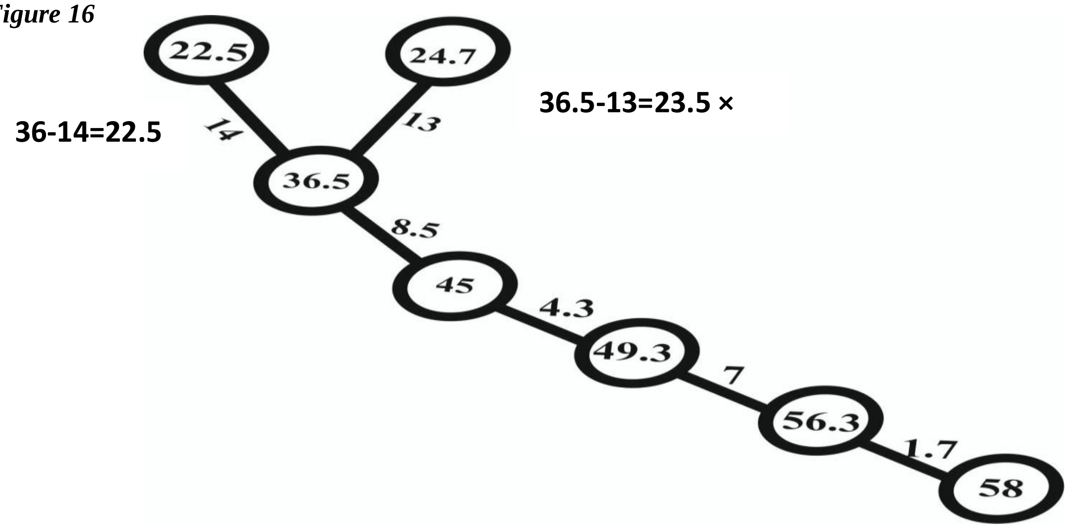

Figure 17

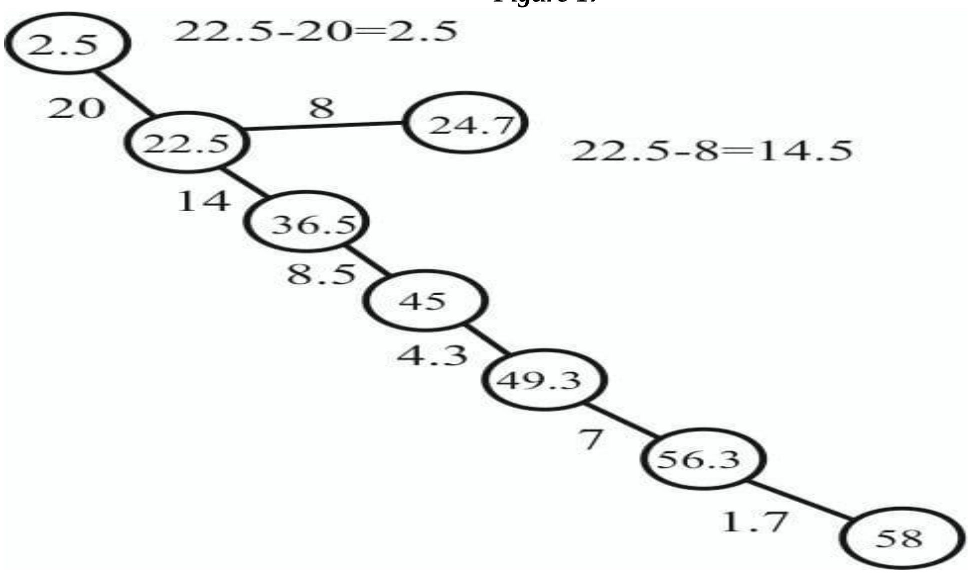

Figure 18

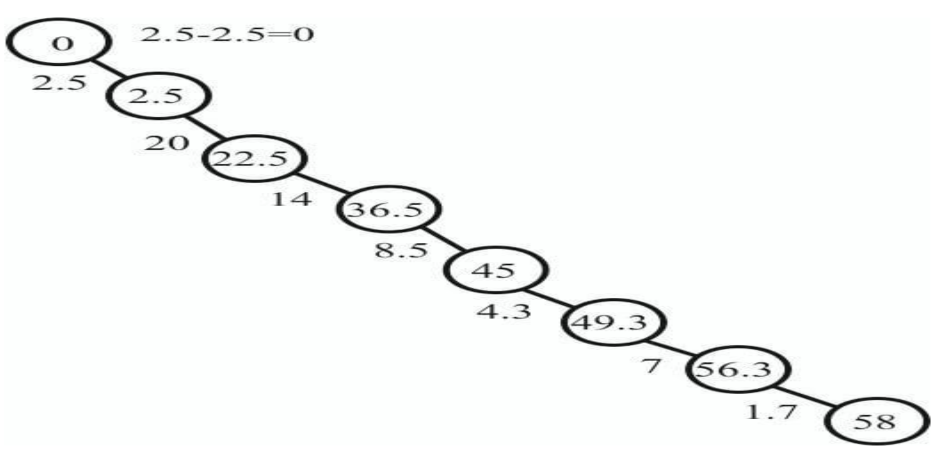

Figure 19

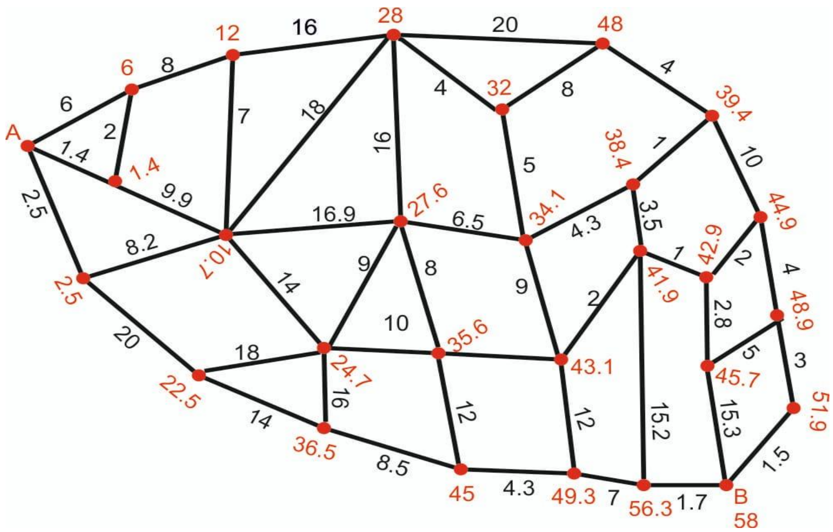

Figure 20

The above figure 20 shows the minimum cost path from place A to B.

IX. APPLICATIONS

Applications of Kruskal’s algorithm are **landing cables, TV Network and Tour operations.**
Example: Most of the cable network companies use the Disjoint Set Union data structure in Kruskal's to find the shortest path to lay cables across a city or group of cities.

X. CONCLUSION

The results obtained from both the algorithms shows that Prim’s Algorithm is very effective to find the shortest distance from vertex A to vertex B. Similarly we obtain a minimum spanning tree by using Kruskal’s Algorithm, in this case we find a MST to reach vertex B to vertex A with minimum cost.

As a result of shortest distance we have obtained a minimum distance of 58 kilometers. If the diesel cost is Rs.72 for a truck whose mileage is $7km/ltr$ we get Rs.596.57 as the total cost from Sathyamangalam to Tirupur.

REFERENCE

[1] RameLikaj, AhmetShala and MirlindBruqi, (2013), Application of Graph Theory to find optimal paths for the transportation problem, Pages 1099-1103, University of Prishtina, Prishtina, Kosovo.
[2] Wen-Chih Chang, Mao-Fan Li, (2008), Learning Kruskal’s Algorithm, Prim’s Algorithm and Dijkstra’s Algorithm by Board Game,Pages 275-284, Springer-Verlag Berlin, Heidelberg.
[3] John Clark and Derek Allan Holton (1995)“A First Look At Graph Theory”, published by Allied Publishers Limited, 1995.
[4] San San Maw, Khin Saw Lin, Lin LinNaing (2019) ,“Dijkstra’s Algorithm for Effective Travelling to the Most Famous Destinations in Mynamar” Pages 4-12.
[5] M.Muthulakshmi, M.M.Shanmugapriya(2016), “Shortest Path Algorithm and its Implementation”, IJMTT V3692) pages 82-85
[6] https://www.hackerearth.com/blog/developers/kruskals-minimum-spanning-tree-algorithm-example/
[7] https://www.statisticshowto.datasciencecentral.com/minimum-spanning-tree/