# A New Quick Algorithm for Finding the Minimal Spanning Tree

**Mai Mismar**

*Al-Quds Open University, Nablus-Palestine*

**ABSTRACT**

*In this paper, I propose a new Algorithm for finding the minimal spanning tree in a graph beginning with any node of the graph. This method is based on creating the cost matrix and then begin to connect the adjacent vertices with the minimum cost using the MinMin criterion. A Minimal Spanning Tree is a subset of the positive edge-weighted undirected graph that connects all the vertices together, without any cycles or loops and with the minimum possible total edge weight. A single graph can have many different spanning trees [1], but all are of the same cost. There are many uses for the minimum spanning trees in our lives, one example is a telecommunications company which is trying to lay out cables in a new neighborhood. So in general a minimal spanning trees are often used in Network design like* telephone, electrical, hydraulic, TV cable, computer, roads.

***Keywords: Graph, cost matrix, Minimal spanning tree, MinMin Criterian.***

## I. INTRODUCTION

Given an undirected and connected network or graph G=(V,E), where V are the vertices or nodes and E are the edges or arcs, then a Spanning Tree of a graph G is a tree that spans G, that is it includes every vertex of G, and is a sub graph of G. Minimum Spanning Tree A minimum spanning tree (MST) of an edge-weighted graph is a spanning tree whose weight (the sum of the weights of its edges) is no larger than the weight of any other spanning tree. Spanning tree is tree that contains each vertex and it doesn't contain any cycle or loop. Agraph contains one or more spanning tree. A Minimum spanning tree (MST) is a spanning tree with weight less than or equal to the weight of every other spanning tree[2]. There may be possibility more than one spanning contain same minimum weight then all the same minimum weighted spanning tree. There are two type of algorithm to solve minimum spanning tree problem: Prim's algorithm and Kruskal's algorithm.

## II. BASIC CONCEPT

***A network**:* is a set of nodes or vertices [4] linked by arcs or edges.

*Directed and Undirected:* an arc is said to be directed if it allows positive flow in one direction and zero flow in opposite direction, otherwise it is undirected. A directed network has its all arcs directed.

***Path***: A sequence of distinct arcs that join two nodes through other nodes regardless of the direction of flow in each arc. A path contains a cycle or a loop if it connects a node to itself through other nodes.

 ISBN: 978-93-86171-60-3

***A connected network:*** if every two distinct nodes are linked by at least one path then it is connected.

***A tree***: is a cycle-free connected network comprised of a subset of all nodes.

***A spanning Tree:*** is a tree that links all the nodes of the network.

***A minimal Spanning Tree:*** is a tree that links all the nodes of the network using the shortest total length of the connecting edges

***Adjacency matrix***: is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent (linked) or not in the graph.

***A spanning Tree Properties:***

Spanning tree: Let G= (V,E) be an un-directed connected graph. A minimal connected sub-graph of G which includes all the vertices of G is a spanning tree of G.

Minimum Spanning Tree: Thus a spanning tree is a minimal sub-graph G1 of G such that [3]:

- V(G) = V(G1) and G1 is connected.
- A connected graph with n vertices must have at least n-1 edges and all connected graphs with n-1 edges are trees.
- If the nodes of G represent cities and the edges represent possible communication links connecting two cities,
- the minimum number of links needed to connect the n cities is n-1.
- The spanning trees of G will represent all feasible choices.
- In practice, the edges will have weights associated with them.
- These might represent the cost of construction, the length of the link etc.
- We are interested in finding a spanning tree of G with minimum cost.

## III. ALGORITHM

- Construct the adjacency matrix filled with the cost on arcs of the graph.
- Beginning with any node of graph do the following:

1- Suppose we choose node I to begin with, then take the submatrix of the adjacency matrix which consists only of the row of node I without its column.

2- Take the minimum cost in that row, suppose it is node j.

3- Linke or mark the arc between the tow nodes I and then delete the row and column of the two nodes.

4- Construct the submatrix which consists of the two rows of the two nodes I and j with the columns of the rest of the other nodes.

5- Take the minimum cost in each row, then take the minimum of minimum. Suppose that MinMin in position (I,k) then link node I with node k and delete k column from the matrix. If you get two MinMin choose one randomly, this means there are more than one spanning tree.

6- So we obtain a submatrix consists of the rows of the linked nodes and the columns of the unlinked nodes, repeat step 5 till you arrive with the last node.

 ISBN: 978-93-86171-60-3

## 1. Examples

***Example (1):*** Consider the network as shown in figure 1, [3] which consists of five nodes given as V= {0, 1, 2, 3, 4} and seven edges.

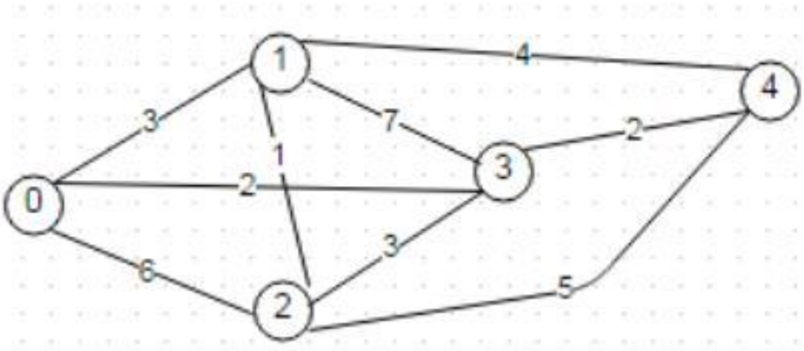



**Figure 1**

And we want to decide the minimal spanning tree in that network.

**Step 1:** we construct the adjacency matrix

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| **0** | | 3 | 6 | 2 | |
| **1** | 3 | | 1 | 7 | 4 |
| **2** | 6 | 1 | | 3 | 5 |
| **3** | 2 | 7 | 3 | | 2 |
| **4** | | 4 | 5 | 2 | |

**Step 2:** Suppose we want to begin with node 0. Then take the sub-matrix of the adjacency matrix which row is the row of the starting node and its columns are the unlinked nodes

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **0** | 3 | 6 | 2 | |

Take the minimum in row, which is 2. That means link node **0** with node **3** with the **cost=2** and marked it on the network, and delete the column of node 3.

**Step 3:** now construct the sub matrix which rows are of the two linked nodes and its columns are the unlinked nodes

| | 1 | 2 | 4 | Min |
|---|---|---|---|---|
| **0** | 3 | 6 | | **3** |

 ISBN: 978-93-86171-60-3

| | | | | |
|---|---|---|---|---|
| **3** | 7 | 3 | 2 | **2** |

**MinMin=2**

This means linking node **3** with node **4** with the **cost=2,** then delete column 2**.**

**Step 4:** now construct the sub matrix which rows are of the linked nodes and its columns are the unlinked nodes

| | **1** | **2** | **Min** |
|---|---|---|---|
| **0** | 3 | 6 | **3** |
| **3** | 7 | 3 | **3** |
| **4** | 4 | 5 | **4** |

**Min Min=3**

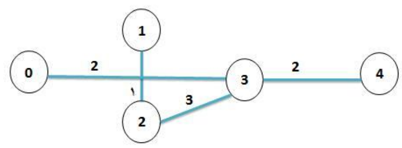
Thus, we have more than one minimal spanning tree, suppose we choose node2, connect node **2** with node **3** with the **cost=3**, then delete column 2.

**Step 5:** now construct the sub matrix which rows are of the linked nodes and its columns are the unlinked nodes.

| | **1** | **Min** |
|---|---|---|
| **0** | 3 | **3** |
| **2** | 1 | **1** |
| **3** | 7 | **7** |
| **4** | 4 | **4** |

**MinMin=1**

So link node **2** with node **1** with the **cost=1**, and delete column 1. Thus, all the nodes are now connected, and the total cost is the sum of the cost =2+2+3+1=8. We obtain this **MST**

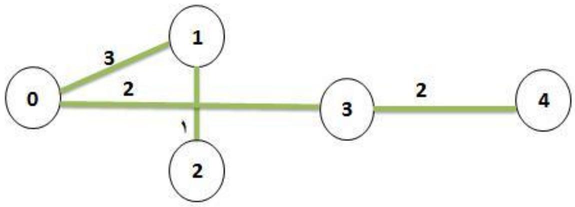



***Note***: let us take the other choice in step 4. I mean if we choose node 1, thus we link node **0** with node **1** with the **cost=3**, then delete column 1.

| | **2** | **Min** |
|---|---|---|
| **0** | 6 | |
| **3** | 3 | **5** |
| **4** | 5 | **2** |
| **1** | 1 | **4** |

**MinMin=1**

So link node 2 with node 1 with the cost=1, and delete column 1. Thus, all the nodes are now connected, and the total cost is the sum of the cost =2+2+3+1=8, but with a different spanning tree with the same cost. We obtain this **MST**




***Example (2):*** Consider the network as shown in figure 2, [2] which consists of four nodes given as V= { 1, 2, 3, 4} and six edges.

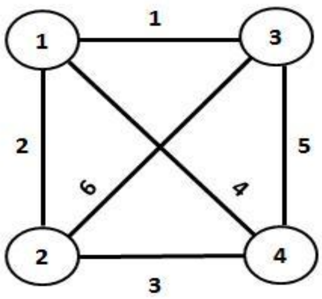



**Figure 2**

**Step 1**: we construct the adjacency matrix.

| | **1** | **2** | **3** | **4** |
|---|---|---|---|---|
| **1** | | 2 | 1 | 4 |
| **2** | 2 | | 6 | 3 |
| **3** | 1 | 6 | | 5 |
| **4** | 4 | 3 | 5 | |

**Step 2:** Suppose we want to begin with node **4**. Then take the sub-matrix of the adjacency matrix as follows:

| | **1** | **2** | **3** |
|---|---|---|---|
| **4** | 4 | 3 | 5 |

Min of (4,3,5)=3. That means link node **4** to node **2** with length **3**,delete column 2.

**Step 3:** now construct the sub matrix which rows are of the linked nodes and its columns are the unlinked nodes.

 ISBN: 978-93-86171-60-3

| | **1** | **3** | **Min** |
|---|---|---|---|
| **4** | 4 | 5 | **4** |
| **2** | 2 | 6 | **2** |

**MinMin=2,** So link node **2** with **1** with the length **2**, then delete column 1

**Step 4: Construct** now construct the sub matrix which rows are of the linked nodes and its columns are the unlinked nodes.

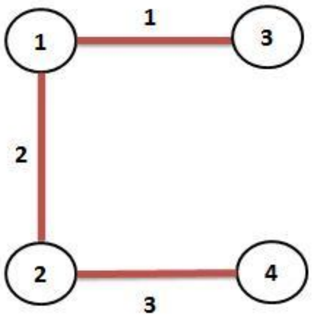
| | **3** | **Min** |
|---|---|---|
| **4** | 5 | **5** |
| **2** | 6 | **6** |
| **1** | 1 | **1** |

**MinMin=1**

Thus, link node **1** to **3** with the length **1**

Thus, all the nodes are now connected, and the total cost is the sum of the cost = 3+2+1=6

We obtain this **MST**




***Example (3)*** Consider the following graph in figure 3 and show the various steps involved in the construction of the minimum cost spanning tree[1] :

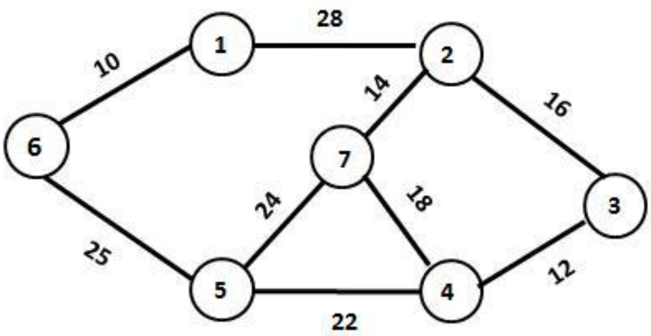



**Figure 3**

**Step 1:** we construct the adjacency matrix

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| **1** | | 28 | | | | 10 | |
| **2** | 28 | | 16 | | | | 14 |
| **3** | | 16 | | 12 | | | |
| **4** | | | 12 | | 22 | | 18 |
| **5** | | | | 22 | | 25 | 24 |
| **6** | 10 | | | | 25 | | |
| **7** | | 14 | | 18 | 24 | | |

**Step 2:** Suppose we want to begin with node 7. Then take the sub-matrix of the adjacency matrix as follows:

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| **7** | | 14 | | 18 | 24 | |

Min=14, thus link node **7** with node **2** with the length=**14**, delete column 2

**Step 3:** now construct the sub matrix which rows are of the two linked nodes and its columns are the unlinked nodes

| | 1 | 3 | 4 | 5 | 6 | Min |
|---|---|---|---|---|---|---|
| **7** | | | 18 | 24 | | **18** |
| **2** | 28 | 16 | | | | **16** |

MinMin=16. So link node **2** with node **3** with the length=**16** and delete column 3

**Step 4:** : now construct the sub matrix which rows are of the three linked nodes and its columns are the unlinked nodes

| | 1 | 4 | 5 | 6 | Min |
|---|---|---|---|---|---|
| **7** | | 18 | 24 | | **18** |
| **2** | 28 | | | | **28** |
| **3** | | 12 | | | **12** |

MinMin =12. So link node **3** with node **4** with the length=**12** and delete column 4.

**Step 5:** : now construct the sub matrix which rows are of the four linked nodes and its columns are the unlinked nodes

| | **1** | **5** | **6** | **Min** |
|---|---|---|---|---|
| **7** | | 24 | | **24** |
| **2** | 28 | | | **28** |
| **3** | | | | |
| **4** | | 22 | | **22** |

MinMin =22. Thus link node **4** with node **5** with the length **22** then delete column 5.

**Step 6:** : now construct the sub matrix which rows are of the five linked nodes and its columns are the unlinked nodes

| | **1** | **6** | **Min** |
|---|---|---|---|
| **7** | | | |
| **2** | 28 | | **28** |
| **3** | | | |
| **4** | | | |
| **5** | | 25 | **25** |

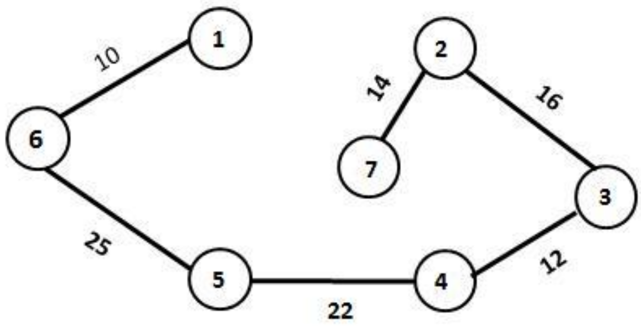
MinMin=25. So link node **5** with node **6** with the length= **25** and delete column 6.

**Step 7:** : now construct the sub matrix which rows are of the six linked nodes and its columns are the unlinked nodes:

| | **1** | **Min** |
|---|---|---|
| **7** | | |
| **2** | 28 | **28** |
| **3** | | |
| **4** | | |
| **5** | | |
| **6** | 10 | **10** |

MinMin =10. So link node **6** with node **1** with the length =**10** and delete column 1. Thus, all the nodes are now connected, and the total cost is the sum of the length = 14+16+12+22+25+10=99.

We obtain this MST:




## IV. CONCLUSION

In this paper I present a new simple efficient algorithm to find the minimum cost spanning tree of an undirected connected weight graph. As there are several application s of minimum spanning tree. Algorithm presented here is based entirely on adjacency matrix and Min of Min criterian. In future I shall concentrate to apply the algorithm for directed graphs.

## REFERENCES

**Journal papers:**

[1] Minimum Cost Spanning Tree using Matrix AlgorithmDr.D vijayalakshmir , R.kalaivani, International Journal of Scientific and Research Publications, Volume 4, Issue 9, September 2014, ISSN 2250-3153.

[2] Survey paper on Different techniques for Minimum Spanning tree Nirav J. Patel, Prof. Shweta, Survey paper on Different techniques for Minimum Spanning tree| ISSN: 2321-9939 .

[3] A Parallel Approach to Solve Minimum Spanning Tree Problem in Network Routing, Saif Ahmed, Jawed Ahmed, International Journal of Emerging Research in Management &Technology ISSN: 2278-9359 (Volume-6, Issue-1), January2017.

**Books:**

[4] Operations Research, An Introduction, Eight Edition, Hamdy A. Taha.