International Journal For Technological Research In Engineering
Volume 1, Issue 9, May-2014 *ISSN (Online): 2347 - 4718*

A NEW APPROACH TO SOLVE MINIMUM SPANNING TREE PROBLEM: MAXIMUM COST PRUNING METHOD

Nirav J. Patel[1] (PG Student), Prof. Manoj Patel[2] (Assistant Professor)
Department of Computer Engineering
Alpha college of Engineering and Technology
Khatraj, Gujarat, India.

Abstract:* *The Minimum Spanning Tree for a given graph is the Spanning Tree of minimum cost for that graph. There are various algorithms to solve Minimum Spanning Tree problem. Existing two algorithms are: Prim's algorithm and Kruskal's algorithm. Time complexity of Prim's and Kruskal's algorithm is O (E lg V). Most of algorithms for Minimum Spanning Tree work only for undirected graph. Our ongoing research will focus on Minimum Spanning Tree problem for both directed and undirected graph.
Index Terms: Spanning tree, Minimum Spanning Tree.

I. INTRODUCTION

A graph G is consist of V and E. V is set vertex and E is edge that connecting two vertex. There are basic two types of graph: **Undirected graph and Directed graph**. Graph can be represented in two ways: Adjacency Matrix and Adjacency List.

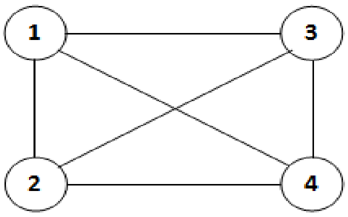
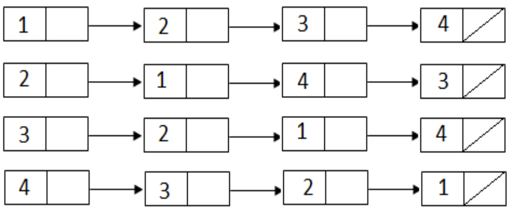


Fig.1.1 [Graph]

This graph can be represented using Adjacency List as follows:



This graph can be represented using Adjacency Matrix as follows:

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Spanning tree of the graph is a tree that contains all the node and doesn't contain cycle.

A **Minimum Spanning Tree (MST)** of a weighted graph is a Spanning Tree in which the sum of the weights of all its edges is minimum of all such possible spanning trees of the graph. The main aim of the Minimum Spanning Tree algorithm is to find the shortest path for given graph in which all nodes will be visited exactly once.

II. MINIMUM SPANNING TREE ALGORITHM

A. Prim's algorithm

In prim's algorithm starting node is selected randomly from given graph. A set which contain all the edges in the graph is created. The edge, which is adjacent to the randomly selected node and with minimum weight among all adjacent edge, is selected from the set and add to new graph if it connects a vertex in the tree with a vertex not in the tree . This process is repeated until every edge in the set connects two vertices in the tree. Time complexity of this algorithm is O(E lg V)

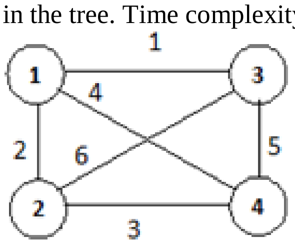


Fig.2.1 [Initial Graph]

Fig.2.2 [Step-1]

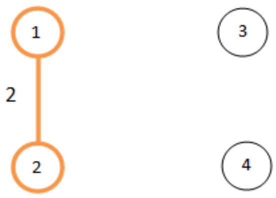


Fig.2.3 [Step-2]

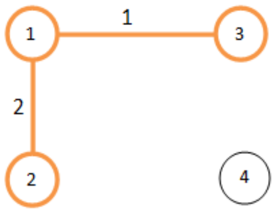


Fig.2.4 [Step-3]

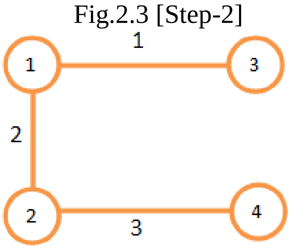


Fig.2.5 [Step-4]

Total Cost of the MST: 2+1+3 = 6

B. Kruskal's algorithm

In kruskal's algorithm all the edges are arranging in increasing order of weight. These edges are placed in a priority queue. Then minimum weighted edge from the queue is selected and placed in to one new forest. This process is repeated until n-1 edges are selected. If addition of any edge in the graph creates cycle then this edge will be rejected. Time complexity of this algorithm is O (E lg V).

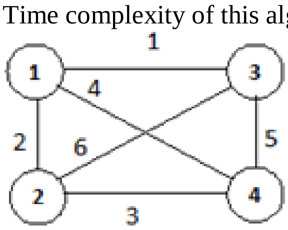


Fig.2.6 [Initial Graph]

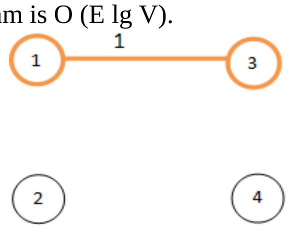


Fig.2.7 [Step-1]

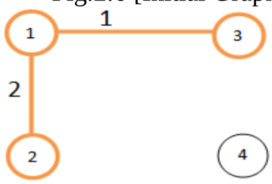


Fig.2.8 [Step-2]

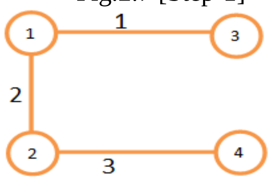


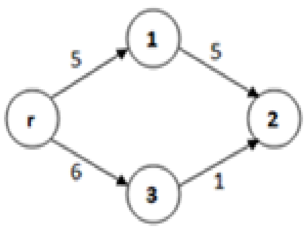
Fig.2.9 [Step-3]

Total Cost of the MST: 1+2+3 = 6

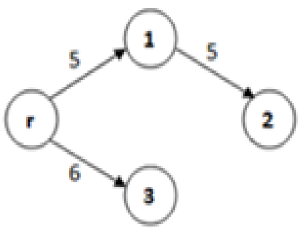
III. PROPOSED WORK

A. Problem Description

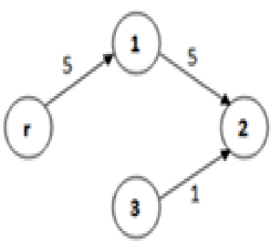
Prim's and Kruskal's algorithm can't give optimal solution in case of directed graph. The proposed algorithm will try to give an optimal solution in case of both directed and undirected connected graph.



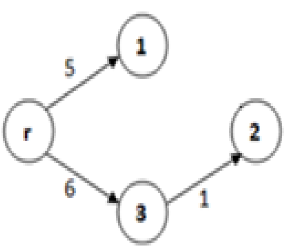
Graph



MST with Prim's algorithm



MST with Kruskal's algorithm



MST with Proposed algorithm

B. Proposed Algorithm

Step-1: Read all the edges E and their weight W of Graph G (V, E) from file.

Step-2: Repeat this step until n-1 edges are present in Graph.
(A) Find out Maximum weighted edge.

(B) Apply following check on this edge in case of directed graph:

- If destination node of this edge has incoming degree greater than 1 and flag is 0 then delete this edge and update weight matrix with weight 0.
- Otherwise update flag for this edge to 1 and go to step (a).

(C) Apply following check on this edge in case of undirected graph:

- Logically delete this maximum weighted edge and apply traverse. If all nodes are visited then delete this edge physically and update weight matrix.
- Otherwise this edge can't be deleted.

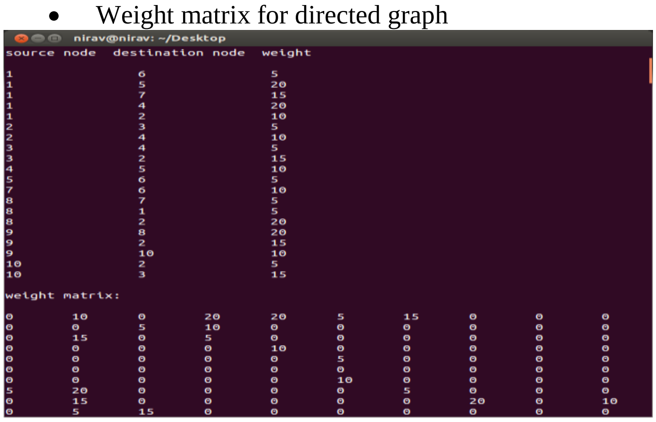
Step-3: Display finally updated weight matrix which is our minimum spanning tree.

Step-4: Finish

IV. IMPLEMENTATION DETAILS

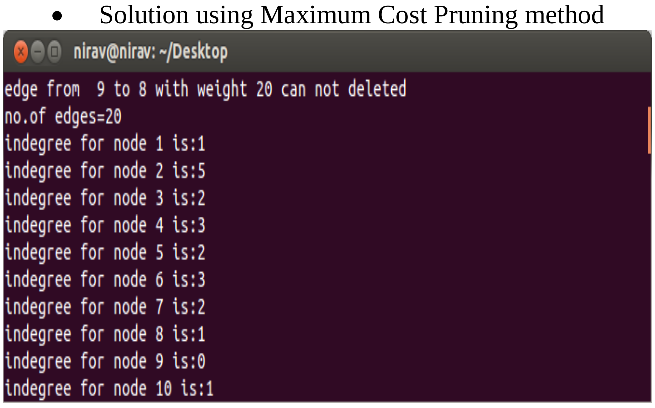
A. Directed Graph

- Weight matrix for directed graph

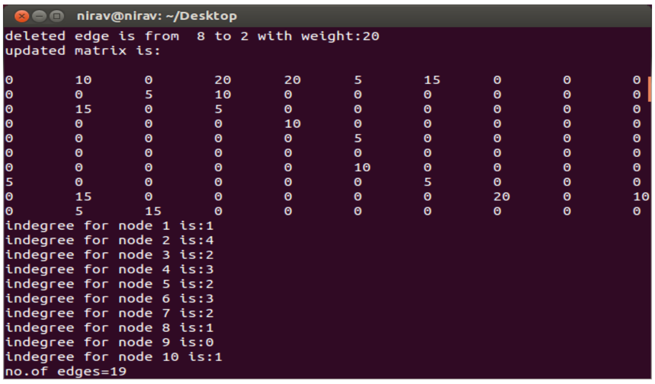


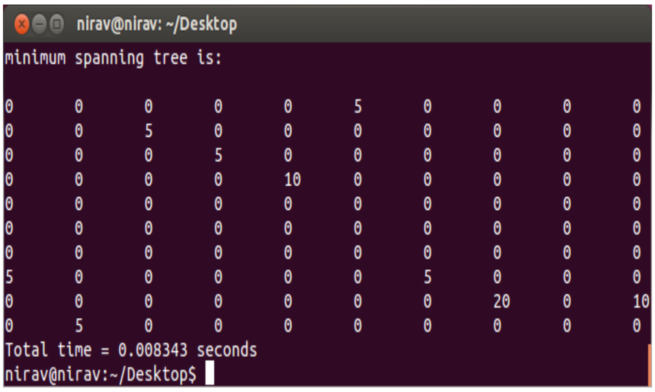
```
nirav@nirav: ~/Desktop
source node   destination node   weight

1             6                  5
1             5                  20
1             7                  15
1             4                  20
1             2                  10
2             3                  5
2             4                  10
3             4                  5
3             2                  15
4             5                  10
5             6                  5
7             6                  10
8             7                  5
8             1                  5
8             2                  20
9             8                  20
9             2                  15
9             10                 10
10            2                  5
10            3                  15

weight matrix:

0    10   0    20   20   5    15   0    0    0
0    0    5    10   0    0    0    0    0    0
0    15   0    5    0    0    0    0    0    0
0    0    0    0    10   0    0    0    0    0
0    0    0    0    0    5    0    0    0    0
0    0    0    0    0    0    0    0    0    0
0    0    0    0    0    10   0    0    0    0
5    20   0    0    0    0    5    0    0    0
0    15   0    0    0    0    0    20   0    10
0    5    15   0    0    0    0    0    0    0
```

- Solution using Maximum Cost Pruning method



```
nirav@nirav: ~/Desktop
edge from  9 to 8 with weight 20 can not deleted
no.of edges=20
indegree for node 1 is:1
indegree for node 2 is:5
indegree for node 3 is:2
indegree for node 4 is:3
indegree for node 5 is:2
indegree for node 6 is:3
indegree for node 7 is:2
indegree for node 8 is:1
indegree for node 9 is:0
indegree for node 10 is:1
```

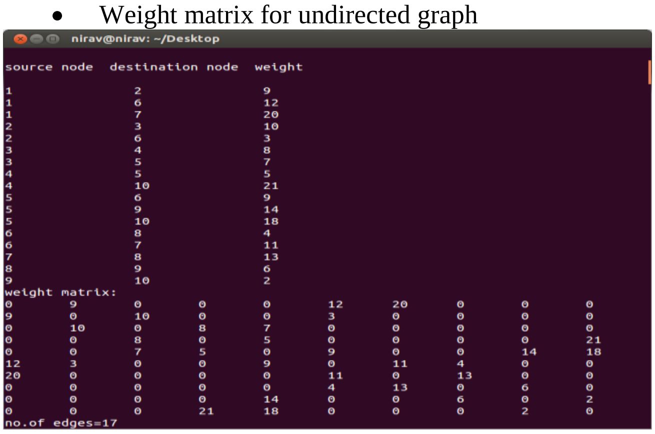





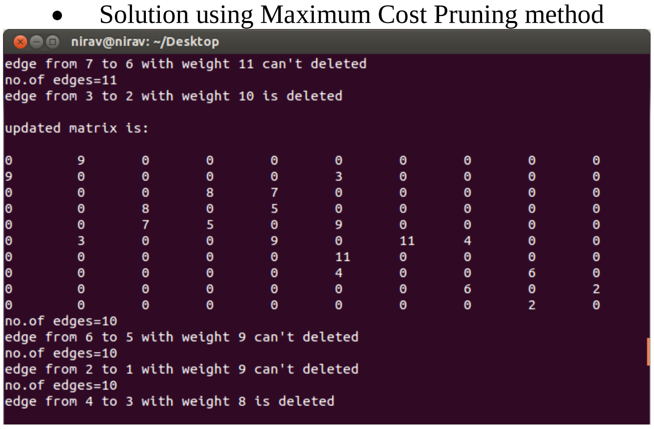
- Existing Prim's and Kruskal's algorithm can't give solution for directed graph

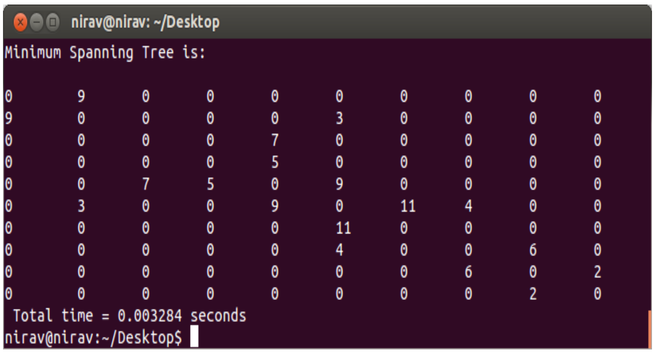
B. Undirected Graph

- Weight matrix for undirected graph



- Solution using Maximum Cost Pruning method





V. CONCLUSION

Here we can conclude that existing algorithm (Prim's and Kruskal's algorithm) cannot give proper solution for Directed Graph. Proposed algorithm (Maximum Cost Pruning method) gives proper solution for both directed and undirected graph.

REFERENCES

[1] International Journal of Engineering and Development Research, Survey paper on different techniques on minimum spanning tree by Nirav Patel and Prof. Shweta Agrawat.
[2] International *Journal* of Computer and Information Technology, Modified Prim's Algorithm, Sunny Dagar
[3] International Journal of Advanced Research in Computer Science And Software Engineering, A New Efficient Technique to Construct a Minimum Spanning Tree, Ardhendu Mandal, Jayanta Dutta and S.C. Pal
[4] IEEE Congress on Evolutionary Computation, An Effective Ant-Based Algorithm for the Degree-Constrained Minimum Spanning Tree Problem, Minh N. Doan
[5] IOSR Journal of Computer Engineering, An Improved Ant-Based Algorithm for Minimum Degree Spanning Tree Problems, Md.Akkas Ali
[6] Parallel Prim's algorithm on dense graphs with a novel extension, Ekaterina Gonina and Laxmikant V. Kal´e, 2007
[7] http://delab.csd.auth.gr/~manolopo/graph/Lec9_MST.pdf
[8] Introduction to Algorithms by Thomas H. Cormen
[9] *www.eecs.berkeley.edu/~egonina/docs/PrimsAlgINTL07.pdf*
[10] http://en.wikipedia.org/wiki/Minimum_spanning_tree
[11] http://en.wikipedia.org/wiki/Prim's_algorithm

