



A New Efficient Technique to Construct a Minimum Spanning Tree

Ardhendu Mandal*

Dept. of Comp. Sc. & Application
University of North Bengal
PO-N.B.U., Dist-Darjeeling
Pin-734013, India

Jayanta Dutta

Dept. of Mathematics
Siliguri Institute of Technology
Siliguri
Pin-734009, India

S.C. Pal

Dept. of Comp. Sc. & Application
University of North Bengal
PO-N.B.U., Dist-Darjeeling
Pin-734013, India

Abstract— A minimum spanning tree of a weighted graph is a tree of the graph which contains all the vertices of the graph and the sum of weights of all its edges are minimum among all such possible trees of the graph. In this paper, we have proposed a new technique and its corresponding algorithm to construct the minimum spanning tree of a weighted graph. This new algorithm is based on the weight matrix of a weighted graph.

Keywords— Graph, Weighted Graph, Tree, Minimum Spanning Tree, Complexity, Sparse Matrix.

I. INTRODUCTION TO MINIMUM SPANNING TREE

A graph is a collection of vertices and edges, and each edge connects a pair of vertices. A tree is an acyclic connected graph. A spanning tree of a graph is a tree that is a sub-graph of the graph and contains all the vertices of the graph. When the edges of a graph are associated with some weights which may represent distance, cost or time etc., it is called a weighted graph. An acyclic weighted graph is a weighted tree. 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. There can be multiple MST of a graph, but all of these MSTs must have unique same total cost.

The problem of constructing MST of an undirected connected weighted graph belongs to the group of classical combinatorial optimization problems, which can be solved by greedy heuristics in polynomial time [1]. In 1926, Boruvka first identifies practical problem related to Minimum Spanning Tree. Recently, there exist several practically relevant variants of MST problem that were proven to be NP-hard. For example the Degree-Constrained MST problem [2], Bounded Diameter MST (BDMST) problem formulated by Nghia and Binh [3], and the Capacitated Minimum Spanning Tree problem [4, 5]. The deterministic MST problem has been well studied and many efficient algorithms have been developed by many researchers [6, 7, 8, 9]. Kruskal's algorithm and Prim's algorithms are the two most commonly used for constructing MST of a graph.

In this paper we present a new technique to construct MST of a weighted graph to construct a minimum spanning tree. The presented technique depends on the weight or cost matrix of the graph.

II. APPLICATION OF MINIMUM SPANNING TREE

Minimum Spanning Tree is fundamental problem with diverse applications. A few of them are mention below:

Network Design: The standard application of MST is in network design i.e. in telephone networks, TV cable network, computer network, road network, islands connection, pipeline network, electrical circuits, utility circuit printing, obtaining an independent set of circuit equations for an electrical network etc.

Clustering Gene Expression Data: A framework for representing a set of multi-dimensional gene expression data as a minimum spanning tree has been proposed by Ying Xu and his associates [10].

Cluster analysis: K clustering problem can be viewed as finding an MST and deleting the k-1 most expensive edges. The k-clustering problem is about dividing a set of objects classify into k coherent groups. The numeric value of weight function specifies "closeness" of two objects.

MST-Based Approximations NP-hard problem: There are several problems in the category of NP-hard. It is yet not possible to find an exact solution of such problems. Minimum Spanning Tree is used in approximating the solution of such NP-hard problems.

Some other Indirect Applications of MST: In addition to the above, there are several other indirect application of MST i.e. Max Bottleneck Paths, Low Density Parity Check (LDPC) codes for error correction, Image registration with Renyi Entropy, Learning salient features for real-time face verification, Reducing data storage in sequencing amino acids in a protein, Model locality of particle interactions in turbulent fluid flows, autoconfig protocol for Ethernet bridging to avoid cycles in a network, in pattern recognition minimal spanning trees can be used to find noisy pixels etc.

III. FORMULATION OF THE PROBLEM

Let, $G = (V, E)$ be an undirected connected weighted graph with n vertices, where V is the set of vertices, E is the set of edges and W be the set of weights associated to respective edges of the graph.

Where,

e_{ij} = The edge adjacent to vertices v_i and v_j .

w_{ij} = The weight associated to the edge e_{ij} .

The Weight Matrix M of the graph G is constructed as follows:

If there is an edge between the vertices v_i to v_j in G then

Set, $M[i,j] = w_{ij}$

Else

Set, $M[i,j] = 0$

For example, the weight matrix of the graph G in Fig. 1 is given below:

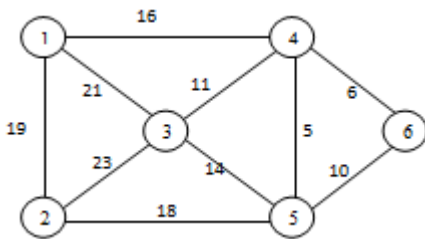


Fig. 1 Graph G1

$$\begin{bmatrix}
 0 & 19 & 21 & 16 & 0 & 0 \\
 19 & 0 & 23 & 0 & 18 & 0 \\
 21 & 23 & 0 & 11 & 15 & 0 \\
 16 & 0 & 11 & 0 & 5 & 6 \\
 0 & 18 & 14 & 5 & 0 & 10 \\
 0 & 0 & 0 & 6 & 10 & 0
 \end{bmatrix}$$

Weight Matrix of G1

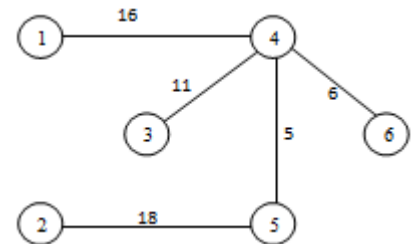


Fig. 2 Minimum Spanning

$$C_{total}(T) = \sum_{i,j=1}^n w_{ij}, \quad \text{where, } w_{ij} \text{ represents the weight assigned to edge } e_{ij} \in T.$$

A
minimu
m
spannin
g
tree

(MST) represents a subgraph T , that contains all the n vertices of G , and has exactly $(n-1)$ edges such that:

For example, Fig.2 shows the Minimum Spanning Tree T of the graph $G1$ (Fig. 1) with a total cost of 56 obtained applying existing well known standard methods.

IV. THE NEW PROPOSED ALGORITHM TO CONSTRUCT MST

The input to the proposed algorithm is the weight matrix M , which consists n^2 elements. But this algorithm only manipulates $n(n-1)/2$ elements those are above the diagonal elements in M i.e. the elements of the upper triangular weight matrix. This algorithm can also be implemented using lower triangular weight matrix.

This proposed algorithm works in the following two passes:

Marking Pass: In this first pass the algorithm marks the candidate edges e_{ij} which will be part of the Minimum Spanning Tree T .

MST Construction Pass: In the second pass, the algorithm constructs the desired Minimum Spanning Tree T including only the marked edges from the upper triangular weight matrix M , which were marked during Marking Pass.

Algorithm:

Input: the weight matrix $M = [w_{ij}]_{n \times n}$ for the undirected weighted graph G .

Output: Minimum Spanning Tree T of G .

Step 1: Start

Step 2: Repeat Step 3 to Step 4 until all $n(n-1)/2$ elements of upper triangular matrix of M are either marked or set to zero or in other words all the nonzero elements are marked.

Step 3: Search the upper triangular weigh matrix M either column-wise or row-wise to find the unmarked nonzero minimum element $M[i,j]$, which is the weight of the corresponding edge e_{ij} in M .

Step 4: If the corresponding edge e_{ij} of selected $M[i,j]$ forms cycle with the already marked elements in the upper triangular elements of the M then

Set $M[i,j] = 0$

Else

Mark $M[i,j]$

Step 5: Construct the graph T including only the marked elements from the upper triangular weight matrix M which shall be the desired MST T of G .

Step 6: Exit.

V. NUMERICAL EXAMPLES

V.1. Six Vertices Example

Consider the following graph G2.

1. Marking Pass:

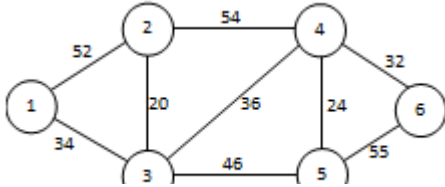


Fig. 3 Graph G2

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

(a) Weight Matrix of G2: Nonzero Minimum Element 20 is marked

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

(b) Next nonzero minimum element 24 is marked

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

(c) Next nonzero minimum element 32 is marked

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

(d) Next Nonzero Minimum Element 34 is marked

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

(e) Next Nonzero Minimum Element 36 is marked. As (n-1) elements are marked, the Marking Pass is over

2. MST Construction Pass:

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

Resultant Weight Matrix after Marking Pass

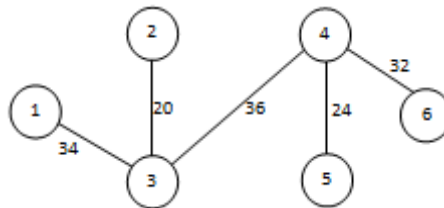
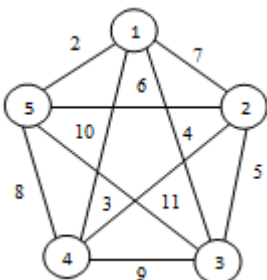


Fig. 4 MST of Graph G2

The Total Cost of the MST = 20 + 24 + 32 + 34 + 36 = 116

V.2. Five Vertices Example

In the following section we consider another weighted graph G3.



$$\begin{bmatrix} 0 & 7 & 4 & 10 & 2 \\ 7 & 0 & 5 & 11 & 6 \\ 4 & 5 & 0 & 9 & 3 \\ 10 & 11 & 9 & 0 & 8 \\ 2 & 6 & 3 & 8 & 0 \end{bmatrix}$$

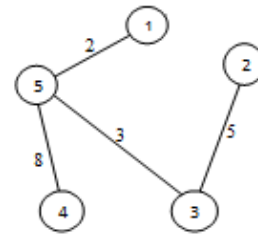
$$\begin{bmatrix} 0 & 7 & 4 & 10 & 2 \\ 7 & 0 & 5 & 11 & 6 \\ 4 & 5 & 0 & 9 & 3 \\ 10 & 11 & 9 & 0 & 8 \\ 2 & 6 & 3 & 8 & 0 \end{bmatrix}$$

Graph G3

Weight Matrix of G3

$$\begin{bmatrix} 0 & 7 & 4 & 10 & 2 \\ 7 & 0 & 5 & 11 & 6 \\ 4 & 5 & 0 & 9 & 3 \\ 10 & 11 & 9 & 0 & 8 \\ 2 & 6 & 3 & 8 & 0 \end{bmatrix}$$

(b) Next Nonzero Minimum Element 3 is marked

$$\begin{bmatrix} 0 & 7 & 4 & 10 & 2 \\ 7 & 0 & 5 & 11 & 6 \\ 4 & 5 & 0 & 9 & 3 \\ 10 & 11 & 9 & 0 & 8 \\ 2 & 6 & 3 & 8 & 0 \end{bmatrix}$$


(c) The next nonzero minimum element 4 creates cycle with the previously marked elements, is reset to 0.

(d) Again both 6 and 8 are reset to 0.

Fig. 6 MST of Graph G3

The Total Cost of the MST = 2 + 3 + 5 + 8 = 18

VI. COMPLEXITY ANALYSIS OF THE ALGORITHM

The input to the algorithm is $n \times n$ weight matrix M, where n is the number of the vertices in the graph under consideration. Out of all $n \times n$ elements, this algorithm only manipulates $n(n-1)/2$ elements of the upper triangular weight matrix M shown below by the shaded background color in Fig. 7. Hence, the algorithm basically operates on a $m \times m$ operational weight matrix, where $m = n-1$, as shown below in Fig. 8.

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 63 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

Fig. 7 $n \times n$ Weight Matrix

$$\begin{bmatrix} 0 & 52 & 34 & 0 & 0 & 0 \\ 52 & 0 & 20 & 54 & 0 & 0 \\ 34 & 20 & 0 & 36 & 46 & 0 \\ 0 & 54 & 36 & 0 & 24 & 32 \\ 0 & 0 & 46 & 24 & 0 & 55 \\ 0 & 0 & 0 & 32 & 55 & 0 \end{bmatrix}$$

Fig. 8 Reducing the $n \times n$ order matrix to order $m \times m$

$$\begin{bmatrix} 52 & 34 & 0 & 0 & 0 \\ 0 & 20 & 54 & 0 & 0 \\ 20 & 0 & 36 & 46 & 0 \\ 54 & 36 & 0 & 24 & 32 \\ 0 & 46 & 24 & 0 & 55 \end{bmatrix}$$

Fig. 9 $m \times m$ Operational Weight Matrix.

Best Case Analysis: The best case is a situation when only the elements either in the first row or first column are marked by searching only the continuous n elements. The complexity for this search will be $O(n)$. Similarly, for constructing the MST with the corresponding marked elements the algorithm will consume $O(n)$ time. Hence, the total best case complexity will be in order of $O(n)$. If we replace n in term of m , the best case time complexity will be in order of $O(m)$, where $m = n-1$.

Worst Case Analysis: The worst case is a situation when all the $n(n-1)/2$ elements of the upper triangular matrix needs to be searched for marking the suitable edges. The complexity for this will be $O(n^2)$. Similarly, for MST construction will have complexity in order of $O(n^2)$. Hence, the total worst case complexity will be in order of $O(n^2)$. If we replace n in term of m , the best case time complexity will be in order of $O(m^2)$, where $m = n-1$.

For large, n the complexity $O(m)$ and $O(m^2)$ are better than $O(n)$ and $O(n^2)$, where $m = n-1$.

VII. CONCLUSION AND FUTURE WORK

The paper presented a new simple and efficient technique to find and construct MST of an undirected connected weighted graph. As there are several applications of Minimum Spanning Tree (MST), this algorithm will be useful to solve such related problems. In future work, we shall concentrate to solve other constrained spanning tree problems using this technique and also try to reduce its' time complexity further.

REFERENCES

[1] Graham, R.L., Hell, Pavol, On the history of the minimum spanning tree problem. Annals of the History of Computing, 7 (1), 44–57, 1985.
 [2] Pagacz, Anna, Raidl, Gunther, Zawislak, Stanisaw, Evolutionary approach to constrained minimum spanning tree problem –commercial software based application. Evolutionary Computation and Global Optimization, 331–341, 2006.
 [3] Nghia, Nguyen Duc, Binh, Huynh Thi Thanh, Heuristic algorithms for solving bounded diameter minimum spanning tree problem and its application to genetic algorithm development, advances in greedy algorithms. Witold Bednorz, 2008.

- [4] Jothi, Raja, Raghavachari, Balaji, Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design. *ACM Transactions on Algorithms* 1 (2), 265–282, 2005.
- [5] Zhou, Gengui, Cao, Zhenyu, Cao, Jian, Meng, Zhiqing, A genetic algorithm approach on capacitated minimum spanning tree problem. *International Conference on Computational Intelligence and Security*, 215–218, 2006.
- [6] Gen M, Cheng R, *Genetic Algorithms and Engineering Optimization*, John Wiley, New York, 2000.
- [7] Oncan T, Design of capacitated minimum spanning tree with uncertain cost and demand parameters, *Information Sciences*, 177(20), 4354-4367, 2007.
- [8] Salazar-Neumann M, The robust minimum spanning tree problem: Compact and convex uncertainty, *Operations Research Letters*, 35(1), 17-22, 2007.
- [9] Zhou G, Gen M, An effective genetic algorithm approach to the quadratic minimum spanning tree problem, *Computers and Operations Research*, 25, 229-247, 1998.
- [10] Y. Xu, V. Olman, and D. Xu. Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees. *Bioinformatics*, 18, 536–545, 2002.