ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 8(29), DOI: 10.17485/ijst/2015/v8i29/84011, November 2015* ISSN (Online) : 0974-5645

Construction of Directed Minimum Cost Spanning Tree using Weight Graph

D. Jasmine Priskilla[1*] and K. Arulanandam[2]

[1]Bharathiar University, Coimbatore – 641046, Tamil Nadu, India; priskillajas@yahoo.com
[2]Department of Computer Applications, GTM College, Gudiyattam – 632604, TamilNadu, India; k.arulanandam@gmail.com

Abstract

Consider a directed weight graph G = (V, E) with n vertices and m edges. A directed minimum cost 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 directed graph. In this paper, I have proposed a new technique and its equivalent algorithm to construct the directed minimum cost spanning tree of a weight graph. This new algorithm is based on the weight matrix of a weighted graph. The input of the proposed algorithm is the weight matrix M, which consist n^2 elements of directed weight graph G. The output of the algorithm is Directed Minimum Cost Spanning Tree. The best case time complexity will be in order of O(m), where m = n-1. The worst case time complexity will be in order of $O(m^2)$, where m = n-1.

Keywords: Directed Graph, Directed Minimum Cost Spanning Tree, Spanning Tree

1. Introduction

A graph is formed by n vertices and m edges connecting the vertices. Creating a graph without having any cycle between vertices is called tree. The subgraph of tree that includes all the vertices of the tree is known as spanning tree. Generally, a graph may have several spanning trees. A minimum cost spanning tree of a weighted graph is formed by its minimum total weight of n-1 edges.

Construction of directed minimum cost spanning tree using weight graph problem belongs to polynomial time algorithm was described in[1–3]. Remaining part of this algorithm referred in Edmond's algorithm[1]. The order of O(min{m logn, n^2}) time is developed by Tarjan[4]. An error in Tarjan's algorithm was repaired by Camerini et. al[5]. By using Fibonacci heaps Gabow et al.[6] improved the running time to O(m+n log n). The improved version of heaps in[7] yielded deterministic O(m log log n) and randomized O(m$\sqrt{}$(log log n)) time. Using weight or cost matrix[8], we present a new technique to construct a directed minimum cost spanning tree.

2. Applications of Directed Minimum Cost Spanning Tree

- Handwriting recognition.
- Hierarchical clustering.
- Computer networks.
- Telecommunications networks.
- Transportation networks.
- Water supply networks.
- Electric grids.
- Circuit design.
- Cluster analysis.

*Author for correspondence

3. Formulation of the Problem

Let directed weight graph G = (V, E), where V is the set of n vertices; E is the set of m edges and W is the set of weights associated from v_i to v_j of the graph.

Let's assume,

e_{ij} = the edge from vertices v_i to v_j.

w_{ij} = The weight of the edge e_{ij}.

From the following rule the weight matrix W of the G is constructed:

If there is an edge from v_i to v_j presented in G then

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

else

Set, W[i,j] = 0

Figure 1 shows the directed weight graph G1.

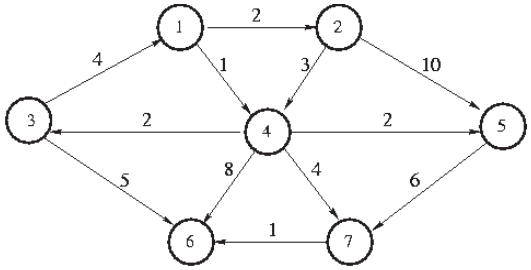

Figure 1. Graph G1.

$$\begin{pmatrix} 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 10 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 2 & 0 & 2 & 8 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

Weight matrix of G1

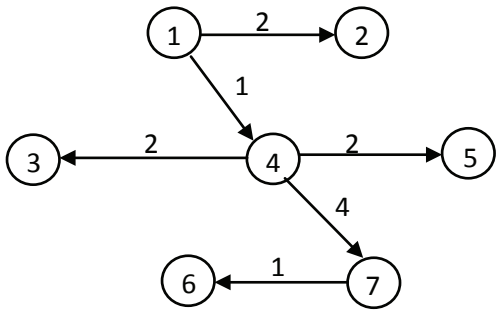

Figure 2. Directed minimum cost spanning tree.

Figure 2 Shows the Directed Minimum Cost Spanning Tree T of the graph G1(Figure 1) with a total cost of 12.

4. Proposed Algorithm to Construct Directed Minimum Cost Spanning Tree

The weight matrix W, consists of n^2 elements are the input to the proposed algorithm.

The following two passes of works are applied for the proposed algorithm.

Pass 1: This pass is called as *marking pass*. In this pass the algorithm marks all the edges e_{ij} which will be part of the directed minimum cost spanning tree T.

Pass 2: This pass is called as *construction pass*. In this pass, the directed minimum cost spanning tree T is constructed from the edges which are selected in pass 1.

Algorithm:

Input: The weight matrix $W=[w_{ij}]$ for the directed weight graph G.

Output: Directed minimum cost spanning tree T of G.

Step 1: Start the Process.

Step 2: Repeat step 3 to step 4 until all the elements of matrix W are either marked or set to zero or (n-1) elements are marked.

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

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

Set W[i, j] = 0

Else

Mark W[i, j]

Step 5: Construct the graph T including only marked elements of directed weight matrix W which shall be the desired directed minimum cost spanning tree T of G.

Step 6: Exit.

5. Example

Consider the following graph G2:

Pass 1: (Marking Pass)

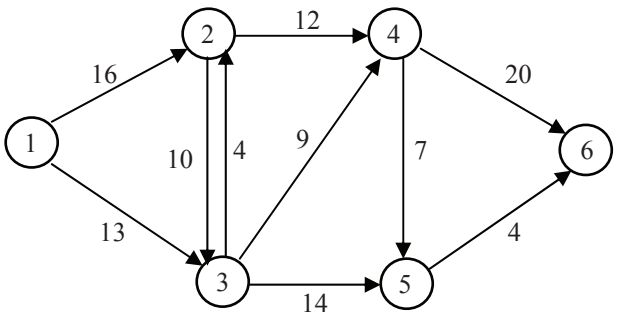

Figure 3. Graph G2.

$$\begin{pmatrix} 0 & 16 & 13 & 0 & 0 & 0 \\ 0 & 0 & 10 & 12 & 0 & 0 \\ 0 & \mathbf{4} & 0 & 9 & 14 & 0 \\ 0 & 0 & 0 & 0 & 7 & 20 \\ 0 & 0 & 0 & 0 & 0 & 4 \end{pmatrix}$$

(a) Weight matrix of G2

Minimum element 4 is marked

$$\begin{pmatrix} 0 & 16 & 13 & 0 & 0 & 0 \\ 0 & 0 & 10 & 12 & 0 & 0 \\ 0 & \mathbf{4} & 0 & 9 & 14 & 0 \\ 0 & 0 & 0 & 0 & 7 & 20 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{4} \end{pmatrix}$$

(b) Next nonzero minimum element 4 is marked

$$\begin{pmatrix} 0 & 16 & 13 & 0 & 0 & 0 \\ 0 & 0 & 10 & 12 & 0 & 0 \\ 0 & \mathbf{4} & 0 & 9 & 14 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{7} & 20 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{4} \end{pmatrix}$$

(c) Next nonzero minimum element 7 is marked

$$\begin{pmatrix} 0 & 16 & 13 & 0 & 0 & 0 \\ 0 & 0 & 10 & 12 & 0 & 0 \\ 0 & \mathbf{4} & 0 & \mathbf{9} & 14 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{7} & 20 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{4} \end{pmatrix}$$

(d) Next nonzero minimum element 9is marked

$$\begin{pmatrix} 0 & 16 & \mathbf{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{4} & 0 & \mathbf{9} & 14 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{7} & 20 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{4} \end{pmatrix}$$

(e) Next nonzero minimum element 10and 12 forms cycle and marked as 0. Next nonzero minimum element 13 is marked. As (n-1) elements are marked, the marking pass is over.

Pass 2: (Construction Pass)

$$\begin{pmatrix} 0 & 16 & \mathbf{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{4} & 0 & \mathbf{9} & 14 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{7} & 20 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{4} \end{pmatrix}$$

The resultant weight matrix after marking pass

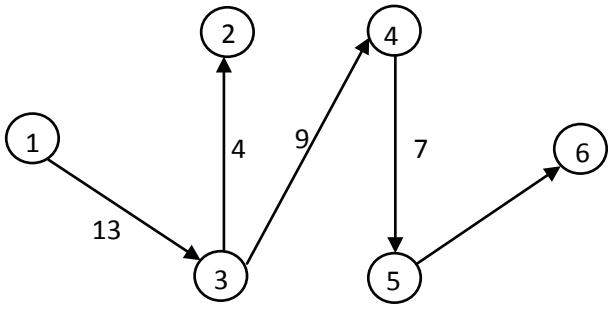

Fig.4. Directed Minimum Cost Spanning Tree of Graph G2

The total cost of the Directed Minimum Cost Spanning Tree = 4 + 4 + 7 + 9 + 13 = 37.

6. Complexity Analysis of the Algorithm

The nxn weight matrix W is the input to the algorithm, where n is the number of the vertices in the graph. It selects n-1 elements to produce T from n^2 elements. Basically it searches n^2-1 elements to produce T. Hence the complexity of this algorithm is $O(n^2)$.

6.1 Best Case Analysis

The situation for the best case is, when, only the elements in first row or first column are available for mark. Other rows or columns are marked as 0. In this situation the complexity for this algorithm will be O(n). When constructing the Directed Minimum Cost Spanning tree with this type of element the algorithm will consume O(n) time. Hence the best case will be O(n) and instead of n, if m replaced, the best case time complexity will be in order of O(m), where m = n-1.

6.2 Worst Case Analysis

The situation for the worst case is, when all the elements in matrix W is consider for searching and marking suitable edges. In this situation the complexity will be $O(n^2)$. When constructing the Directed Minimum Cost Spanning tree with consider all the elements of matrix W, the algorithm will consume $O(n^2)$ time. Hence the best case will be $O(n^2)$ and instead of n, if m replaced, the worst case time complexity will be in order of $O(m^2)$, where m = n-1.

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

Include conclusion

7. References

1. Edmonds J. Optimum branchings. Journal of Research of the National Bureau for Standards. 1967; 71B(4): 125–30.
2. Bock F. An algorithm to construct a minimum spanning tree in a directed network. Developments in Operations Research. Gordon and Breach; 1971. p. 29–44.
3. Chu YJ, Liu TH. On the shortest arborescence of a directed graph. Scientia Sinica. 1965; 14:1396–400.
4. Tarjan RE. Finding optimum branchings. Networks. 1977; 7:25–35.
5. Camerini PM, Fratta L, Maffioli F. A note on finding optimum branching. Networks. 1979; 9:309–12.
6. Gabow HN, Galil Z, Spencer TH, Tarjan RE. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica. 1986; 6:109–22.
7. Mendelson R, Tarjan RE, Thorup M, Zwick U. Melding Priority Queues. Proceedings of SWAT'04. Springer LNCS. 2004; 3111:223–35.
8. Ardhendu, et al. A new efficient technique to construct a minimum spanning tree. International Journal of Advanced Research in Computer Science and Software Engineering. 2012 Oct; 2(10):93–7.