A NEW ALGORITHM FOR DETERMINING THE MINIMUM SPANNING TREE

LI YONG SUN

Computing Center, Jilin University, Changchun, China.

LI SANG HO

Jilin Vocational Teacher's College, Changchun, China.

LIU JI JUN

Computing Center, Jilin University, Changchun, China.

Abstract

This paper presents a new algorithm for determining a minimum spanning tree of a connected weighted graph. This algorithm is simpler, more direct, and high speed.

1. Introduction

One approach to determining a minimum cost spanning tree of a graph has been given by Kruskal. In this approach a minimum cost spanning tree, T, is built edge by edge. As an example, consider the graph of figure 1.1 (a). The edge of this graph are

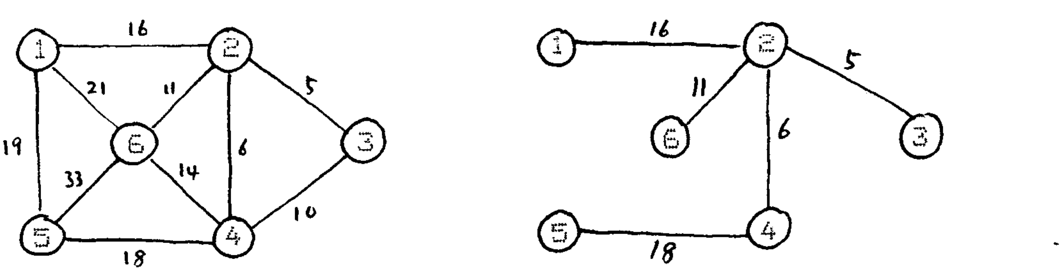

Figure 1.1 Graph and A spanning Tree of Minimum Cost considered for inclusion in the minimum cost spanning tree in the order (2, 3), (2, 4), (3, 4), (2, 6), (4, 6), (1, 2), (4, 5), (1, 5), (1, 6) and (5, 6).

This corresponds to the cost seguence 5, 6, 10, 11, 14, 16, 18, 19, 21 and 33. The first two edges (2,3) and (2,4) are included in T. The next edge to be considered is (4.3). This edge,however,connects two vertices already connected in T ant so it is rejected. The edge (2,6) is selected while (4,6) is rejected as the vertices 4 and 6 are already connected in T and the inclusion of (4,6) would result in a cycle . Finally, edges (1,2) and (4,5) are included. The spanning obtained (figure 1.1 (b)) has cost 56. It is somewhat surprising that this straightforward approach should always result in a minimum spanning tree. For clarity, the Kruskal algorithm is written out more formally in figure 1.2. Initially, E is the set of all edges in G. the only functions we wish to perform on this set are:(i) derternining an edge with minimum cost (line 3), and (ii) deleting that edge (line 4).

```
1.  T<--Ø
2.  While T contains less than n-1 edges do
3.     choose an edge (v,w) from E of lowest cost;
4.     delete (v,w) from E;
5.     if (v,w) does not create a cycle in T
6.        then add (v,w) to T
7.        else discard (v,w)
8.     end
```

Figure 1.2 Early Form of Minimum spanning Tree Algorithm-Kruskal.

2. Fundamental Subgraph

We ehall make the following definition.

Definition 2.1. Let G=(V,E) be a connected weighted graph, Where $V=(V_1,\dots,V_n)$. Choose an edge with minimum weight from the edges connecting vertex Vi,it is called the basic, and denoted e'_i (i=1,...,n). Let $E'=(e'_1,\dots,e'_n)$,then G1=(V,E') is called the fundamental subgraph of G.

Proposition 2.1. Let G=(V,E) be a conneted weighted graph, where $V=(V_1,\dots,V_n)$,then G=(V,E') has the properties:

(1) Any of the cycly is not formed in the G1;

(2) Any of the connected components of G1 all is tree;

(3) Suppoe G1 have K connected commponents, then

$$K = n - \mathrm{Card}(E')\,.$$

where the card (E') denotes the cardinal number of the set E'.

Proof. Suppose G1 have one cycle, then this cycle has an edge with maximum weight. But either verter of this edge has 2 basic edges. This contradicts the definition 2.1. Hence, there is no such cycle and (1) is proved.

According to the (1), the proof of the (2) is straightforward.

Therefore E' has at least a pair. frepeated members, because G has the edge with minimum weight. Therefore we many suppose that the E' has N pair of repeated members:

$$e'_1,\ e'_2,\ \ldots\ldots,\ e'_{2K-1},\ e'_{2K}$$

where $e'_{2\alpha-1} = e'_{2\alpha}$, for $\alpha = 1, \ldots, k$

But $G((V1, W1), e'), \ldots\ldots, G((Vk, Wk), e')$

are disjoint connected components. Where

$$e'_{2\alpha-1} = (V_\alpha\ , W_\alpha\ ), \qquad \text{for} \quad \alpha = 1, 2, \ldots\ldots, N$$

According to the definition 2.1, we may prove that any of the vertices in $G((V_\alpha, W_\alpha), e'_{2\alpha-1})$ is not connected with the any of the one in $G((V_\beta, W_\beta), e')$, for $\alpha \neq \beta$.

And we may also prove that how many vertices in $V \setminus (V_1, W_1, \ldots, V_K, W_K)$ are not form the connected componet. In case this is not so, then a connected component is formed. Hence in this connected component there must be an edge with minimum weight. Consequently, we obtain that another repeated member in E'. This contradicts the assumption on E'. Hence, the proposition is proved.

3. Algorithm DCA

Algorithm DCA Given a connected weighted graph G=(V,E). Suppose Gi be a subgraph of G and it has Ki connected components.

Algorithm DCA of a subgraph Gi proceeds as follows:

Step 1. If Ki=1, then end;

Step 2. If Ki>1, then delete the edges connecting any two vertices of every conneted components from E;

Step 3. Choose an edge with minimum weight from remaining edzgs;

Step 4. Add this edge to Gi so obtain Gi+1.

Algorithm MST Let G=(V,E) be a connected weighted graph. Suppose G1 be the fundamental subgraph with K connected components of G.

Step 1. Proceed algorithm DCA of G1;

Step 2. Proceed algorithm DCA of G2;

... ...

Step k-1. Proceed algorithm DCA of Gk-1.

Theorem 3.1 The algorithm MST generates a minimum spanning tree Gk for G.

Proof. The proof may follow from the definition 2.1 with the algorithm DCA.

Example 3.1 Determine the minimum spanning tree of graph of figure 1.1 (a).

We may obtain the fundamental subgraph of this graph(figure 1.1 (b)).Since K=1,therefore this subgraph is the minmum spanning tree.

Example 3.2 Determine the minimum spanning tree for the connected graph defined by distance matrix:

$$D = \begin{bmatrix} 0.0 \\ 0.9 & 0.0 \\ 0.1 & 0.9 & 0.0 \\ 0.7 & 0.7 & 0.5 & 0.0 \\ 0.2 & 0.9 & 0.4 & 0.5 & 0.0 \\ 0.9 & 0.9 & 0.8 & 0.9 & 0.8 & 0.0 \\ 0.8 & 0.9 & 0.8 & 0.9 & 0.8 & 0.3 & 0.0 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.7 & 0.6 & 0.0 \end{bmatrix}$$

LET G=(V,E) denote the graph defined by matrix D,then we may obtain the fundamental subgraph G1 of G (figure 3.1):

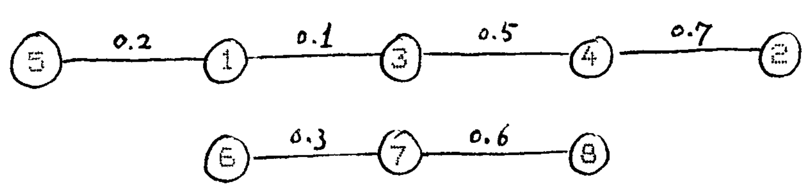

Figure 3.1 the fundamental subgrap.

It is thus clear that this subgraph has two connected components. Therefore delete the edges connecting any two vertices of every connected components from E, then may obtain the set of remaining edges:

S= {(1,6),(1,7),(1,8),(2,6),(2,7),(2,8),(3,6),(3,7),(3,8),
(4,6),(4,7),(4,8),(5,6),(5,7),(5,8)}

Now choose an edge (3,6) with minimum weight 0.8 from S (naturally may choose any edge in {(3,6),(5,6),(1,7),(3,7),(5,7) }

Finally and this edge (3,6) to the graph of figure 3.1 we obtain following graph:

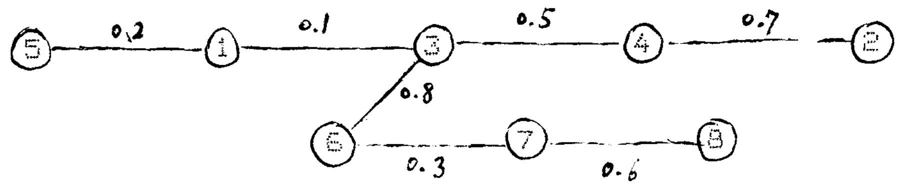

Figure 3.2 A minimum spanning tree .

This straightforward approach should always result in a minimum spanning tree.

References

(1) Kruskal, J.B., On the shortest spanning subtree of a graph and the traveling salesman problem, Proc. Amer. Math. Soc., Vol.7, No.1, P 48-50, (1956).

(2) Wang Xiang-hao, Guan Ji-wen and Liu Xu-hua, Discrete mathematics ,(1983).

(3) Ellis Horowitz and sartaj Sahni, Fundamental of dada structures, Computer Science Press, INC, P296-301, (1976).

(4) Li Yong sun and Li Sang Ho, WK-Clustering algorithm. (in press).

(5) Fang Kaitai and Pan Enpei ,Clustering analysis, (1982).