International Journal of Pure and Applied Mathematics
Volume 106 No. 8 2016, 67-72
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v106i8.10
ijpam.eu

SOLVING SPANNING TREE PROBLEMS IN A CONNECTED WEIGHTED SIMPLE GRAPH

S. Kalidasan[1], P. Pandian[2]
[1,2]Department of Mathematics
School of Advanced Sciences
VIT University, Vellore-14, INDIA

Abstract: In this paper, a new algorithm namely, spread search algorithm is proposed for finding a maximum/minimum spanning tree of a given weighted connected simple graph. The proposed algorithm differs from the existing algorithms and it has the time complexity of $O((m-k)^2)$ where m is the number of edges and k is the number of pendant edges in the given graph. Numerical examples are presented for illustrating the solution procedure of the spread search algorithm.

AMS Subject Classification: 68R10, 05C05, 05C85
Key Words: graph, weighted graph, maximumn spanning tree, spread search algorithm

1. Introduction

Graph theory [1, 10] began with Leonhard Euler in his study of the Bridges of Konigsberg problem. Currently, it is one of the most important areas of applied mathematics. Many problems of practical interest can be represented by graphs. In general, graph theory has a wide range of applications in diverse fields. In reality, graph theory is a branch of Combinatorics, but it is cross-disciplinary between Mathematics, Science, Technology and Optimization theory.

Network which is a connected graph has been studied by many researchers recently because it has real life applications in the areas like as telecommunication, wireless sensor networks, transporting plan etc. The main objective in

Received: February 15, 2016
Published: March 28, 2016

the design process in networks is to reach total connectivity at optimum cost / distance / profit / stable. Many researchers have contributed more related to connectivity problems in the sense of theoretical as well as in computational. Minimum spanning tree problem in a connected graph is one of most important in combinatorial optimization and has important applications in transportation, communications, distribution systems, etc. The minimum spanning tree problem has been well studied and many efficient algorithms have been developed by Dijkstra [4], Kruskal [6], Prim [9], Gabow et al. [5], Bondy and Murty [1], and Christofides [2]. Among these, the two basic algorithms namely, Kruskals algorithm and Prims algorithm are greedy algorithms which are more popular for solving minimum (maximum) spanning tree in a connected graph. The maximum capacity root problem is solved by maximum spanning tree using modified prims algorithm [9].

In this paper, we propose a new algorithm namely, spread search algorithm for solving maximum spanning tree problems in weighted connected simple graphs which totally differes from the existing algorithms [3, 7, 8]. The proposed algorithm has the time complexity $O((m-k)^2)$ where m is the number of edges and k is the number of pendant edges in the given graph. The mathematical proof of the spread search algorithm for a maximum spanning tree of a given graph is provided. The solution procedure of the proposed algorithm is demonstrated with numerical example.

2. A Weighted Connected Simple Graph

A graph G is an ordered triple $G = (V(G), E(G), I_G)$, where $V(G)$ is a non empty set, $E(G)$ is a set disjoint from $V(G)$, and I_G is an "incidence" map that associates with each element of $E(G)$, an unordered pair of elements of $V(G)$. A simple graph is a graph which has neither loops nor multiple edges. A graph G in which every edge is assigned a real number is called a weighted graph.

A connected acyclic graph is called a tree. A spanning tree of a graph G is a spanning subgraph, that is a tree. A weighted graph is a graph G in which each edge e has been assigned a non-negative number $w(e)$, called the weight of e. The weight of a spanning tree, T is the sum of the weights of the edges in the tree T. A maximum (minimum) spanning tree of G is a spanning tree of G with maximum (minimum) weight.

Now, we define the following new terms in a weighted graph.

Definition 1. The node u is said to be a neighbour of the node x if $w(x, u) \neq 0$

Definition 2. The node u is said to be a strong neighbour of the node x if $w(x, u)$is the maximum of $\{w(x, u) : u \in V is\ a\ neighbour\ of\ x\}$

Definition 3. The node u is said to be a weak neighbor of the node x if $w(x, u)$is the minimum of $\{w(x, u) : u \in V is\ a\ neighbour\ of\ x\}$

3. Spread Search Algorithm

Now,we propose the following new algorithm namely, spread search algorithm for finding a maximum spanning tree of a given weighted connected simple graph.

The proposed algorithm proceeds as follows:

Algorithm: Let $G = (V, E)$ be a connected weighted simple graph with n vertices and m edges

Step 1: Collect all the pendent edges of G and form a set. Let it be S.

Step 2: If $|S| = n-1$ stop the computation and $T_0 = (V, S)$ is the maximum spanning tree. If not, move to the Step 3..

Step 3: Find an edge $e_1 = (u_1, u_2)$ in $G_1 = G - S$ such that $W(e_1) = W(u_1, u_2)$ is maximum. If more than one occur, select any one edge.

Step 4: Construct $T_1 = T_0 + e_1 = (V, S_1)$ where$S_1 = S \cup e_1$. If$|S_1|$ $= n - 1$,stop the computation and T_1 is the maximum spanning tree. If not, move to the Step 5..

Step 5: Find an edge e_2 whose one end vertex is an end vertex of the edge e_1 in the graph $G_2 = G_1 - e_1$ such that $W(e)$ is the maximum of $\{W(u, u_r), u_r$ is a strong neighbour of a non-isolated end vertex u of the edge e_1 in the graph $G_2\}$.

Step 6: Construct $T_2 = T_1 + e_2 = (V, S_2)$ where $S_2 = S_1 \cup e_2$. If $|S_2|$ $= n - 1$, stop the computation T_2 and is the maximum spanning tree. If not, move to the Step 7..

Step 7: Find an edge e_3 whose one end vertex is an end vertex of the edge e_1 or e_2 in the graph$G_3 = G_2 - e_2$ such that $W(e_3)$ is the maximum of $\{W(u, u_r), u_r$ is a neighbour of a non-isolated end vertex of the edge or in the graph $G_2\}$. with $T_2 + e_3$ not containing a cycle. If e_3 forms a cycle delete edge from G_3 continue the step 7.

Step 8: Construct $T_3 = T_2 + e_3 = (V, S_3)$ where $S_3 = S_2 \cup e_3$. If $|S_3|$ $= n - 1$, stop the computation and T_3 is the maximum spanning tree. If not, move to the Step 7..

Step 9: Continue the Step 7. and the Step 8. for $G_4 = G_3 - e_2$ and its reduced graphs.

Theorem 4. *In a connected weighted simple graph G, the spread search algorithm provides a maximum weighted spanning tree of G.*

Proof. Let T_0 be a graph obtained by the spread search algorithm. Now, in the proposed algorithm, an edge could not be selected if it forms a cycle. Therefore, T_0 is a tree in G. Now, since G is connected, each vertex in G has atleast one strong neighbour in G. Now, since the proposed algorithm moves from one vertex to its strong neighbour, each vertex in G is connected in T_0. Therefore, T_0 is a spanning tree of G.

Claim. *To prove that T_0 is a maximum spanning tree of G.*

That is, $W(T_0) > W(T)$, for all spanning tree T of G. Let T be a spanning tree of G. Suppose $W(T_0) < W(T)$.

This implies that there exist an edge $e = (u, v)$ in T, but $e \notin T_0$ such that the weight of the edge $e = (u, v)$ is greater than the weight of the path connecting the vertices u and v in T_0. Now, since $e \notin T_0$, u is not a strong neighbour of v in G and there exist strong neighbours for u and v in G, u_1 and v_1 respectively such that (u, u_1) and (v, v_1) are in T_0. Since the strong neighbour for u in G isu_1 and the strong neighbour for v in G is v_1, we have $W(u, u_1) > W(u, v)$ and $W(v, v_1) > W(u, v)$. Now, $W(u, u_1) > W(u, v)$ and $W(v, v_1) > W(u, v)$, the weight of the path connecting u and v containing (u, u_1)or (v, v_1) is greater than the weight of the edge (u, v) which is contradicts to the hypothesis $W(T_0) < W(T)$. Therefore, $W(T_0) > W(T)$, for all spanning tree T of G. Thus, T_0 is a maximum spanning tree of G. Hence, the theorem is proved. □

Now, we present the following numerical example for understanding the solution procedure of the spread algorithm.

Example 3.1. Consider the following weighted connected simple graph G.

Claim. *To determine a maximum spanning tree of the given graph.*

Now, since and by Step 1. and Step 2., we have $S = (v_2, v_8)$, $T_0 = (V, S)$.

Now, since $W(v_7, v_3)$ is maximum in $G - (v_2, v_8) = (v_7, v_8)$ and by Step 3. and Step 4, we have $T_1 = T_0 + (v_7, v_3)$.

Now, by the Step 5. and the Step 6., we have $T_2 = T_1 + (v_3, v_4)$.

Now, repeat the Step 9.,

We have $T_7 = T_8 + (v_6, v_2)$ and we stop the computation since the number of edges in T_7 is 7.

Now, by Step 8., T_7 is a maximum spanning tree of the given graph which is given below and its total weight is 126.

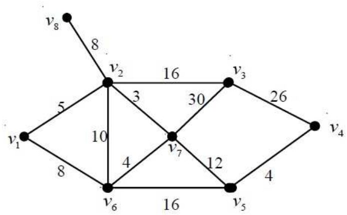

Figure 1: G

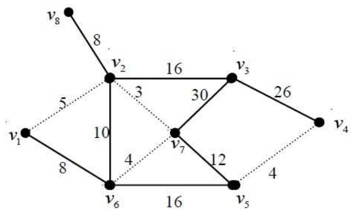

Figure 2: T_7

4. Conclusion

In this paper, we consider a weighted connected simple graph. Spread search algorithm is proposed for solving maximum spanning tree problems in weighted connected simple graph. The time complexity of the proposed algorithm is $O((m-k)^2)$ where m is the number of edges and k is the number of pendent edges. The proposed algorithm is based on the newly defined concept namely, strong neigbhour and weak neighbour of a vertex in the graph and, it totally differs all other existing algorithms.

References

[1] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, Macmillan and Elsevier, New York, 1976.

[2] N. Christofides, *Graph Theory: An Algorithmic Approach*, Academic Press, New York, 1975.

[3] Cuneyt F.Bazlamacci,Khalil S.Hindi, Minimum-weight spanning tree algorithms A survey and empirical study, *Computers Operations Research*, **28**, (2001), 767-785.

[4] E.W. Dijkstra, A note on two problems in connection with graphs, *Numerische Mathematik*, **1**(1959), 269-271.

[5] H.N.GaBow, Z.Galil, T.Spencer and R.E.Tarjan, Efficient algorithms for finding minimum spanning trees in undirected and directed graph, *Combinatorial*, **6**(1986), 109-122.

[6] J.B. Kruskal Jr., On the shortest spanning subtree of a graph and traveling salesman problem, *Proceeding of AMS*, **7** (1956), 45-50.

[7] Nimesh Patel and Patel, A survey on: Enhancement of minimum spanning tree, *International Journal of Engineering Research and Applications*, **5** (2015), 06-10.

[8] Patel Nirav Jasavantbhai and Shweta Agrawat, Survey paper on different techniques for minimum spanning tree, *International Journal of Engineering Development and Research*, **1**(2014), 22-25.

[9] R.C. Prim, Shortest connection networks and some generalizations, *Bell System Technical Journal*, **3**6 (1957), 1389-1401.

[10] C.Vasudev, *Graph theory with Applications*, New Age International Publishers, New Delhi, 2006.