RESEARCH ARTICLE OPEN ACCESS

A Survey on: Enhancement of Minimum Spanning Tree

Nimesh Patel*, Dr. K. M. Patel**
*(Department of Computer Engineering, R K University at Rajkot, INDIA)
** (Department of Computer Engineering, R K University at Rajkot, INDIA)

ABSTRACT
Minimum spanning tree can be obtained for connected weighted edges with no negative weight using classical algorithms such as Boruvka's, Prim's and Kruskal. This paper presents a survey on the classical and the more recent algorithms with different techniques. This survey paper also contains comparisons of MST algorithm and their advantages and disadvantages.
***Keywords* –** Cord, DWCM, FCM, Graph, LC-MST, MST, Tree

I. INTRODUCTION

A Minimum Spanning Tree of a weighted graph is a spanning tree in which the sum of the weight of all its edges is a minimum of all such possible spanning tree of the graph. Minimum spanning Tree must be finding from the Graph. A collection of vertices and edges makes a graph, and each edge connects a pair of vertices [1, 2, 3, 4,].

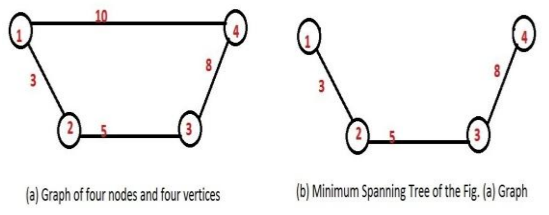
10
1
4
3
8
2
5
3
(a) Graph of four nodes and four vertices
1
4
3
8
2
5
3
(b) Minimum Spanning Tree of the Fig. (a) Graph

Fig.1 graph and its MST

There are two types of graph, Directed graph and undirected graph. A directed graph is graph in which a set of vertices are connected together, where all the edges are directed from one vertex to another. A directed graph is also known as digraph or a directed network. In contrast, a graph where the edges are bidirectional is called an undirected graph. In the directed graph edges have a direction associated with them. An undirected graph is one in which edges have no orientation. The edge (a, b) is identical to the edge (b, a).The maximum number of edges in an undirected graph without a self-loop is n (n - 1)/2 [10].

1.1 Application of MST

1.1. Applications of MST are used in the design of computer and communication networks, telephone networks, links road network, islands connection, pipeline network, electrical circuits, utility circuit printing, obtaining an independent set of circuit equations for an electrical network, etc.

1.1.2 It offers a method of solution to other problems to which it applies less directly, such as network reliability, clustering and classification problems.
1.1.3 Used to find the approximation solution for the NP hard problems.

1.2 Objective of MST

1.2.1 To minimize cost of the tree spanning tree for both directed and undirected.
1.2.2 To minimize load on the network.
1.2.3 To eliminate the cycle from the graph from the MST.
1.2.4 To improve the complexity of the MST.

II. MST CLASSICAL ALGORITHM

There are various classical algorithms available which describe below.

Kruskal's , Prim's and Boruvka's algorithm is a greedy algorithm which used to find a minimum spanning tree for a connected weighted undirected graph. This means when the total weight of all the edges is minimized in the tree, at that time it finds a subset of the edges which forms a tree which includes every vertex

2.1 Kruskal algorithm:

This algorithm first appeared in Proceedings of the American Mathematical Society during 1956, and was written by Joseph Kruskal. In Kruskal's algorithm all edges are shorted in non decreasing order and selected the lowest edges first for becoming a minimum spanning tree. If there is a cycle generated during the implantation then selected edges will be removed from the graph and next lowest edges are selected. This will repeat till (n-1) edges will be added in to the graph. Using simple data structure Kruskal's algorithm complexity is O (E log E) time, or equivalently, O (E log V) time. Where E is the number of edges in the graph and V is the number of vertices [1, 2, 21,].

Advantages are:

1) Easy to understand

2) Give good result for large number of vertices and edges.

Disadvantages are:

1) Difficulty of checking whether arcs form cycles makes it slow and hard to program

2) Same weight may increase the complexity.

2.2 Prim's algorithm:

The algorithm was developed in 1930 by Jarnik and later rediscovered by computer scientist Robert C. Prim in 1957 and then again rediscovered by Edsger Dijkstra in 1959. Therefore it is also known as the DJP (Dijkstra-Jarnik Problem) algorithm, the Jarnik algorithm, or the Prim–Jarnik algorithm. Using a simple binary heap data structure complexity is O(|E| log |V|) where |E| is the number of edges and |V| is the number of vertices. Using Fibonacci heap in dense graph complexity is O(|E| + |V| log |V|), which is asymptotically faster [2, 9, 12, 16].prim's algorithm steps are given below:

1) Create a set MST Set that keeps track of vertices already included in MST.

2) Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign key value as 0 for the first vertex so that it is picked first.

3) While MST set doesn't include all vertices

a) Pick a vertex u which is not there in MST Set and has minimum key value.

b) Include u to MST Set.

c) Update key value of all adjacent vertices of u. To update the key values, iterate through all adjacent vertices. For every adjacent vertex v, if weight of edge u-v is less than the previous key value of v, update the key value as weight of u-v.

Advantages are:

1) Easy to understand.

2) Root node is selected so clear about the starting node.

Disadvantages are:

1) Time taken to check for smallest weight arc makes it slow for large numbers of nodes.

2) Difficult to program, though it can be programmed in matrix form

3) Same weight may increase the complexity when one of the weights is eliminated in a cycle

2.3 Boruvka's algorithm:

It was first published by Otakar Boruvka in 1926. It is a method of constructing an efficient electricity network. This algorithm was again discovered by Choquet in 1938, then it rediscovered by Florek, Lukasiewicz, Perkal, Steinhaus, and Zubrzycki in 1951, then again discovered by Sollin in 1965. Sollin was the only computer scientist in this list living in an English speaking country. So, this algorithm is frequently called Sollin's algorithm. The algorithm starts visiting each vertex and adding the cheapest edge from that vertex to another vertex in the graph, if edges already added in the graph then in will neglected. It will continue joining these edges until all vertices is visited in spanning tree. Boruvka's algorithm taken O (log V) iterations of the outer loop until it terminates. Therefore it take O (E log V) time to run. Where E is the number of edges, and V is the number of vertices in graph [12, 18].

Advantages are:

1) If the edge costs are distinct, or are made distinct by using a tie-breaking rule then Boruvka's algorithm can be serialized into a specialization of the generic algorithm.

Disadvantages are:

1) It is complicated to implement without serialization.

2.4 Karger, Klein Tarjan:

Use random sampling in combination with linear time algorithm for verifying spanning tree. This computational model is unit cost random access with restriction. This operation allowed on edges weights for binary comparison. It runs in O(m) time with high probability in restricted random access model. This algorithm proposed by Karger whose time complexity is O(n log n + m)[12, 17]. The O (m) time complexity is due to Klein and Tarjan follow two properties. 1) Cycle property: For any cycle in the graph, if weight of an edge of cycle is larger than the weights of all other edges of cycle, then this edge cannot belong to an MST. 2) Cut property: For any cut cycle in the graph, if the weight of an edge of cycle is strictly smaller than the weights of all other edges of cycle, then this edge belongs to all MST of the graph.

Advantages are:

1) It's allowed on edges weights for binary comparison.

Disadvantages are:

1) Random sampling in combination with linear time algorithm gives restriction for unit cost while access randomly.

2.4.1 Cycle property:

For any cycle in the graph, if weight of an edge of cycle is larger than the weights of all other edges of cycle, then this edge cannot belong to an MST.

2.4.2 Cut property:

For any cut cycle in the graph, if the weight of an edge of cycle is strictly smaller than the weights of all other edges of cycle, then this edge belongs to all MST of the graph.

III. LITERATURE SURVEY

In this section, lot of research works has been recorded from past few years. They are presented here:

3.1 Combinatorial algorithm [5]:

This algorithm is based on Difference Weighted Circuit Matrix. To find a minimum spanning tree for a connected weighted graph with no negative weight can be obtained using classical algorithms such as Prim's and Kruskal. Both of two give the single minimum spanning tree. It sometimes needs to generate the second minimum spanning tree, third, fourth and so on. This algorithm performs two major tasks. 1) Cord: The edges that are not in the spanning tree of a graph are called the chord. That is the sub graph S is the collection of Chord of the graph G with respect to S the Spanning tree of the graph. 2) DWCM: The abbreviation is Difference Weighted Circuit Matrix. It is the little bit of modification of the FCM. A sub matrix in which all rows correspond to a set of fundamental circuits is called a Fundamental circuit matrix. If n is the number of vertices and e is the number of edges in a connected graph, then the matrix is an (e-n-1) *(n-1) matrix. Here the branch weights are present on the column head as branch mark. The chords (e-n-1) are for the row representation. Here each cell of the matrix is assigned difference weight of the chord and the branches participating for generating circuit. When the column head presented, this chord is joined to the spanning tree.

Advantages are:

Sometimes in real life that minimal path can't be reached due to some circumstances, in that case the next minimal spanning tree is useful.

Disadvantages are:

1) Complexity is more because of generating more than one spanning tree more.

3.2 Euclidean based MST algorithm [6]:

It is based on the well separated pair decomposition. This is introduced by Callahan and Kosaraju.

There are two standard techniques for implementing adding edge process: one is the Union-find data structure, another one is the labeling method. Here structure wrapped with the tree map method. In which each index of the labeling maps with integer key type data for the sequence of vertices in a cycle. At the time of execution loop adding a new edge into the minimum spanning tree, it checks that if both vertices have not been added to any of the cycle yet. If it wasn't added then it create a new key. Then it maps this integer value with the new cycle. In that case one vertex has been added to one of the cycles with other vertex, it just add this new vertex into the cycle containing the other vertex. Another situation is that both vertices have been added to the same cycle.

In this case edge is redundant for the new minimum spanning tree. Through this algorithm storage space and running time efficiency is improve. To compute the EMST of n points in the space, one can link each pair of edges.

Advantages are:

1) It greatly improves the storage space and running time efficiency over traditional approaches.

Disadvantages are:

1) Hard to implement

3.3 LC-MST algorithm [7]:

Here, least-cost minimum spanning tree (LC-MST) problem is defined as a method to construct a minimum cost spanning tree that has the least-cost edges in the network by using the distance or cost matrix. The method presents a new algorithm based on the distance matrix to solve the LC-MST problem.LC-MST algorithm steps are given below.

1. Input the distance matrix D = [dij] nxn for the weighted graph G (V, E), where V is the set of vertices and E is the set of edges.

2. For all i, j find the least-cost element in each column j and set the other elements to zero.

3. Construct the preferred link matrix (PLM) by using step 2.

4. Construct the nodes-set matrix (NSM) by using PLM matrix constructed in step 3.

5. Combine the node-pairs in step 4 to construct the candidate spanning tree.

6. If there are any duplicating node-pairs, keep one of them. If there is a set of node-pairs, construct a cycle, remove the one that has the largest cost.

7. Output the least-cost minimum spanning tree.

Time complexity is less than the DC-MST and CMST algorithms. Still complexity is O(n^2).

Advantages are:

1) Simple and efficient method to solve the LC-MST problem in less time.

LC-MST time complexity is less than the time complexity of both the DC-MST and CMST algorithms.

Disadvantages are:

1) Complexity is higher O(n^2).

3.4 Heap base bucket sorting algorithm [8]:

In this algorithm, edges are in group and place in to the bucket. Then select the minimal edges from the bucket and put in to the heap. If there is a cycle generated then removes those edges from the bucket and then next minimal edges will select. This will repeat till n-1 edges will find out from the heap. A clear cut method to form the buckets is to link the elements which describing the edges. In the same bucket to form a linear list and to use an array of list heads which point to the front of the lists. Note that this is not the only possible storage organization. The method of math sort through also possible to arrange the edges by making a chain of exchange operations. Complexity for sorting edge is improving in O(m) instead of O(m log m) [5]. It will work effectively in case of uniformly distributed random number for wide class.

Advantages are:

1) This algorithm is very fast if the edge costs are from a distribution which is close to uniform.

Disadvantages are:

1) Complexity is O(m log m) in worst case occurs when there is a strong peak in the distribution of the edge costs.

3.5 Modified prim's algorithm [9]:

In this algorithm, Instead of choosing randomly, root node is chosen with minimum edge weight. Remaining procedure is same as used by prims. Due to this only minimum weight edges are included. Although complexity remains same as prim's algorithm.

Advantages are:

It gives slightly better performance in case where minimum weight edge is required from the starting phase of minimum spanning tree formation.

Disadvantages are:

Complexity is remaining same.

3.6 Visit, Mark and Construct MST algorithm [10]:

In this algorithm, adjacency matrix is used which help to reduce the step at the time of constructing MST. This method is based on the kruskal algorithm with modification which used improve the complexity of the MST algorithm for the undirected graph. This method is purely for the undirected graph. So the weight of the 1 to 2 vertices is same for the 2 to 1 vertices. See the below Fig.2. In which edges 1 to 2 contain 52 weights and the weight for the edges 2 to 1 is also 52.So, it is same for undirected graph.

$$\begin{bmatrix} 0 & 52 & 10 & 16 \\ 52 & 0 & 9 & 5 \\ 10 & 9 & 0 & 2 \\ 16 & 5 & 2 & 0 \end{bmatrix}$$

Fig.2 n*n weighted matrix

Now for the vertices 1 to 1, 2 to 2....n to n. there is no weight and if there is a weight then it automatically removed because of generating a cycle. So, it places 0 as infinite. Now if we have n vertices then we have n*n adjacency matrix. So, need to perform n^2 steps, because it will check all the elements from the graph. So, author first removes unused row column from the adjacency matrix. Here first row and last column are never used during the implementation because edges 1 to 2 has the same 2 to 1 and edges 1 to 1 is always 0 or automatically eliminated because of generating cycle. So, adjacency matrix has (n-1) rows and (n-1) columns. See the below adjacency Fig.3 and Fig.4.

$$\begin{bmatrix} 0 & 52 & 10 & 16 \\ 52 & 0 & 9 & 5 \\ 10 & 9 & 0 & 2 \\ 16 & 5 & 2 & 0 \end{bmatrix}$$

Fig.3 Reducing the n*n order matrix to order m*m where m = (n-1)

$$\begin{bmatrix} 52 & 10 & 16 \\ 0 & 9 & 5 \\ 9 & 0 & 2 \end{bmatrix}$$

Fig.4 m*m operational weight Matrix

So, n^2− 2n +1 steps will be performed. So, complexity is O (m^2) where m is (n-1) [7]. This algorithm works in the following two passes. 1) Mark Phase: In which algorithm marks the candidate edge from the graph for the minimum spanning tree. 2) MST Construction Phase: In the second phase, 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.

In this algorithm, minimum weight are marked and visited first. Once weight is visited and it doesn't create a cycle then it will be added to the list of minimum spanning tree edges. Otherwise it will be removed and next minimum weight should be taken for the further procedure. This will happen till n-1 edges get for the minimum spanning tree edges. Once the n-1 edges are get than it will stop algorithm and calculate total cost.

Advantages are:

Complexity is O(n) for the best case.

Disadvantages are:

Complexity is O(n^2) for the best case.

IV. CONCLUSION

This survey paper presents classical algorithms and advance MST algorithm & it is observed that complexity is very high because of cycle in the graph and the edges with the same weight. It also observed that complexity can be improved using following steps. 1) Marked and visit the maximum weight of the edges. 2) If it creates a cycle then eliminate those edges from the current graph. 3) Above two steps will be repeated till edges = vertices-1.

References

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein,

"*Introduction to Algorithms (3rd Ed.)*", (PHI Learning Private Ltd.-2009), 624-642.

[2] Mehlhorn, Kurt, and Peter Sanders. *Algorithms and data structures: The basic toolbox*, (Springer, 2008)

[3] Wu, Bang Ye, and Kun-Mao Chao. "*Spanning trees and optimization problems*", (CRC Press, 2004)

[4] Motwani, Rajeev, and Prabhakar Raghavan, Randomized algorithms, (Chapman & Hall/CRC, 2010)

[5] Barun Biswas, Krishnendu Basuli, Saptarshi Naskar, Saomya Chakraborti and Samar Sen Sarma, "*A combinatorial algorithm to generate all spanning trees of a weighted graph in order of increasing cost*", *CoRR, September 2012.*

[6] Chaojun Li, "*Euclidean Minimum Spanning Trees Based on Well Separated Pair Decompositions*", *Dave Mount, May 22, 2014.*

[7] M.R. Hassan, "*An efficient method to solve least-cost minimum spanning tree (LC-MST) problem*", *Journal of King Saud University - Computer and Information Sciences (2012) 24, 101-105.*

[8] Katajainen, Olli Nevalainen and Jyrki, "*An alternative for the implementation of Kruskal's minimal spanning tree algorithm*", *Science of Computer Programming 3.2 (1983), 205-216.*

[9] Sunny Dagar, "*Modified Prim's Algorithm*", *IJCIT, ISSN 2218-5224 (ONLINE), VOLUME 03, ISSUE 02, 2012.*

[10] Mandal, Ardhendu, Jayanta Dutta, and S. C. Pal. "*A New Efficient Technique to Construct a Minimum Spanning Tree*", *International Journal 2.10 (2012).*

[11] J.B Kruskal. *On the shortest spanning subtree of a graph and traveling salesman problem, Proc.Amer.Math.Soc.7 1956, pp.48-50*

[12] Eisner, Jason. "*State of the art algorithms for minimum spanning trees a tutorial discussion*", (1997).

[13] GaBow, H. N., Galil, Z., Spencer, T., Tarjan, R. E, "*Efficient Algorithms for Finding Minimum Spanning Trees in Undirected and Directed Graph*", *Combinatorial - 6(2), 109-122 (1986).*

[14] 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.*

[15] Jothi, Raghavachari, Balaji, Raja, "*Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design*", *ACM Transactions on Algorithms 1 (2), 265–282, 2005.*

[16] Karger, Robert E. Tarjan, Philip N. Klein, and David R. , "*A randomized linear-time algorithm to find minimum spanning trees*", *Journal of the ACM-42.2 - (1995), 321-328.*

[17] Chazelle, Bernard, "*A minimum spanning tree algorithm with inverse-Ackermann type complexity*", *Journal of the ACM - 47.6 – (2000), 1028-1047.*

[18] Nesetril, Jaroslav, Helena Nesetrilova and Eva Milkovs, "*Otakar Boruvka on minimum spanning tree problem translation of both the 1926 papers, comments, history*", *Discrete Mathematics 233.1, (2001),3-36.*

[19] Wang, Xiaochun, Xiali Wang, and D. Mitch Wilkes, "*A divide-and-conquer approach for minimum spanning tree-based clustering*", *Knowledge and Data Engineering, IEEE Transactions on 21.7, (2009), 945-958.*

[20] F Bazlamacci, Khalil S Hindi and Cuneyt, "*Minimum-weight spanning tree algorithms a survey and empirical study*", *Computers & Operations Research 28.8, (2001), 767-785.*

[21] Pettie, Seth, and Vijaya Ramachandran, "*An optimal minimum spanning tree algorithm*", *Automata, Languages and Programming, Springer Berlin Heidelberg, 2000, 49-60*