Survey paper on Different techniques for Minimum Spanning tree

[1]Nirav J. Patel, [2]Prof. Shweta Agrawat
[1]PG Student, [2]Assistant Professor
Department of Computer Engineering, Alpha college of Engineering and Technology,
Khatraj, Gujarat, India.
[1]niravpatel470@gmail.com, [2]shweta.agravat@gmail.com

***Abstract*— For a given graph there are number of vertices and edges. Each edge has weight associated with it. Minimum spanning tree is the tree with lowest weight and contain all vertices in which there is no any cycle. There are different algorithms to solve the problem of minimum spanning tree. In this paper we have discuss about prim's algorithm, kruskal's algorithm, modified prim's algorithm etc. There are different ideas in different algorithms to find minimum spanning tree.**

***Index Terms*— Graph, Spanning tree, Minimum Spanning Tree, Prim's algorithm, Kruskal's algorithm. :**

I. INTRODUCTION

A graph G is consist of V and E. V is set vertex and E is edge that connecting two vertex. There are basic two types of graph: Undirected graph and Directed graph. There are two ways to represent the graph:

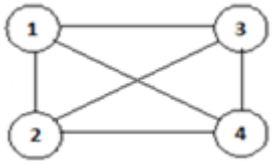


Fig.1.1 [Graph]

Adjacency list:

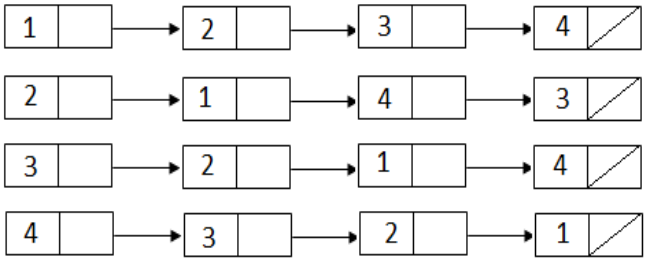


Fig.1.2 [Adjacency list representation]

Adjacency Matrix:

	1	2	3	4
1	0	1	1	1
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0

Fig.1.2 [Adjacency Matrix representation]

Adjacency list use when the graph is sparse graph and Adjacency Matrix use when the Graph is Dense graph. In sparse graph |E| is much less then $|V|^2$ [7]. In dense graph |E| is close to $|V|^2$ [7].There are two type of algorithm to search the graph: Breath first search (BFS) And Depth first search (DFS). Spanning tree is tree that contains each vertex and it doesn't contain any cycle. A graph contains one or more spanning tree. A Minimum spanning tree (MST) is a spanning tree with weight less than or equal to the weight of every other spanning tree[10]. There may be possibility more than one spanning contain same minimum weight then all the same minimum weighted spanning tree consider as a minimum spanning tree. If each edge have distinct weight then there will be only one unique minimum spanning tree[10]. There are two type of algorithm to solve minimum spanning tree problem: Prim's algorithm and Kruskal's algorithm.

II. BASIC CONCEPT

A. *Graph Terminology*

Adjacent: If two nodes are connected with a one common edge then it's called adjacent node.

Path: A sequence of vertex that connects two nodes is called path.

Cycle: In Graph the starting node and ending node is equal then it's called cycle.

Simple Graph: A Graph which does not contain any loop is called Simple Graph.

Connected Graph: The graph in which there is direct or indirect path from each vertex to every other vertex is called connected graph.

Sub Graph: A graph which has subset of edges and subset of vertices of the original graph is called sub graph.

Complete Graph: The graph in which each vertex is directly connected with every other vertex is called complete Graph.

Directed Graph: the edges in a graph have a direction is called direction graph.

Undirected Graph: The edge in a graph which does not have a direction is called undirected graph.

In-degree: The number of edges coming inside the node is called In-degree of that particular node.

Out-degree: The number of edges going outside the node is called Out-degree of that particular node.

B. *Minimum spanning tree Algorithm*

1) Prim's algorithm

In prim's algorithm starting node is selected randomly from given graph. A set which contain all the edges in the graph is created. The edge, which is adjacent to the randomly selected node and with minimum weight among all adjacent edge, is selected from the set and add to new graph if it connects a vertex in the tree with a vertex not in the tree . This process is repeated until every edge in the set connects two vertices in the tree. Time complexity of this algorithm is O(E lg V)

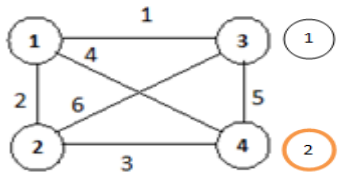


Fig.2.1 [Initial Graph]

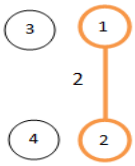


Fig.2.2 [Step-1]

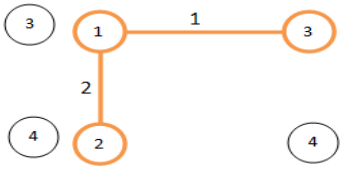


Fig.2.3 [Step-2]

Fig.2.4 [Step-3]

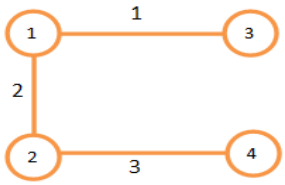


Fig.2.5 [Step-4]

Total Cost of the MST: 2+1+3 = 6

2) Kruskal's algorithm

In kruskal's algorithm all the edges are arranging in increasing order of weight. These edges are placed in a priority queue. Then minimum weighted edge from the queue is selected and placed in to one new forest. This process is repeated until n-1 edges are selected. If addition of any edge in the graph creates cycle then this edge will be rejected. Time complexity of this algorithm is O(E lg V).

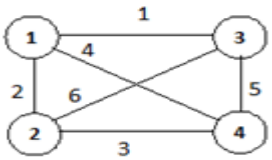


Fig.2.6 [Initial Graph]

Fig.2.7 [Step-1]

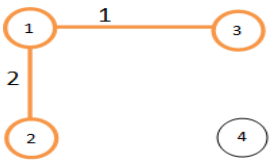


Fig.2.8 [Step-2]

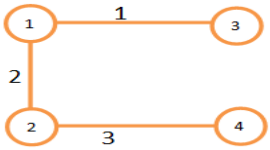


Fig.2.9 [Step-3]

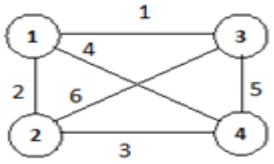
Total Cost of the MST: 1+2+3 = 6

3) *Modified Prim's algorithm*

This algorithm is modified version of the prim's algorithm. In Prim's algorithm chooses a root node randomly and starts processing but in modified prim's algorithm minimum weighted edge is selected first[1]. This algorithm is slightly different than prim's algorithm. In modified prim's algorithm select minimum weighted edge so it gives slightly better performance than original prim's algorithm[1]. Time complexity of this algorithm is same as prim's algorithm.

4) *A New Efficient Technique to Construct a Minimum Spanning Tree*

In this algorithm there is used weight matrix to solve minimum spanning tree problem. This algorithm there are two passes: Marking pass and Construction pass. In marking pass the algorithm select candidate edge (minimum weighted edge) and marked it[2]. The marked edge will be one part of the minimum spanning tree. In the construction pass construct minimum spanning tree which include marked edge which is mark during marking pass. Time complexity in best case is O(n) and time complexity in worst case is $O(n^2)$. This technique is only used for undirected graph.



$$\begin{bmatrix} 0 & 2 & 1 & 4 \\ 2 & 0 & 6 & 3 \\ 1 & 6 & 0 & 5 \\ 4 & 3 & 5 & 0 \end{bmatrix}$$

Fig.2.10 [Weighted Graph] Fig.2.11 [Weight matrix for Graph]

$$\begin{bmatrix} 0 & 2 & \mathbf{1} & 4 \\ 2 & 0 & 6 & 3 \\ 1 & 6 & 0 & 5 \\ 4 & 3 & 5 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & \mathbf{2} & \mathbf{1} & 4 \\ 2 & 0 & 6 & 3 \\ 1 & 6 & 0 & 5 \\ 4 & 3 & 5 & 0 \end{bmatrix}$$

Fig.2.12 [Marking pass-1] Fig.2.13 [Marking pass-2]

$$\begin{bmatrix} 0 & \mathbf{2} & \mathbf{1} & 4 \\ 2 & 0 & 6 & \mathbf{3} \\ 1 & 6 & 0 & 5 \\ 4 & 3 & 5 & 0 \end{bmatrix}$$

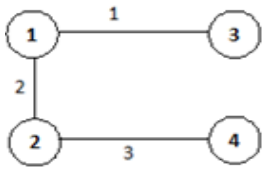


Fig.2.14 [Marking pass-3] Fig.2.15 [Final Solution]

Total cost of the MST: 1+2+3 = 6

5) *An Effective Ant-Based Algorithm for the Degree-Constrained Minimum Spanning Tree Problem*

In minimum spanning tree there no node in the spanning has the degree more than a specified integer d, is called Degree Constrained Minimum Spanning Tree (DCMST).Finding the degree-constrained minimum spanning tree of a graph is a well-studied NP-hard problem[3].

There are six steps in this algorithm: Initialization, Exploration, Solution Construction, Local Optimization, Global Pheromone Update and Stopping Criteria.

Initialization stage: Calculate initial value to prepare for the next stage.

Exploration: Generate candidate edges require to construct good solution in the next stage[3]. Each ants has its own list this list contain all unvisited nodes in the graph[3]. The ants move to the next vertex based on following:

- Traverse on small weighted edge and utilize information from the others.

Solution construction: there are two methods.

a) All the edges arrange in decreasing order according its performance then apply kruskal's algorithm.
b) n edges will be selected from set and then these edges are sorted in increasing order of weight then these edges are sorted in increasing order of weight then apply kruskal's algorithm.

Local Optimization: It uses cut and paste method. In cut and paste method a new spanning tree formed by picking vertex i and deleting edge ij so there will create two disconnected component[3]. These two components are reconnecting by using another vertex such that total edge weight becomes smaller [3].

Global Pheromone update: When any Update are created from any vertex then it updated globally.

Stopping Criteria: The algorithm stops if no improvement found in consecutive cycle and it has run for maximum cycle.

This algorithm improved the previous ant-based algorithm by adding the local optimization, two methods of constructing solutions and modifying the probability function of state transition rule.

6) *An Improved Ant-Based Algorithm for Minimum Degree Spanning Tree Problems*

A spanning tree of a connected graph is a sub graph, with least number of edges that still spans. The problem of finding degree constraint spanning tree is known to be NP-hard. In this algorithm ant-Based algorithm for finding minimum degree spanning trees (MDST) and gives improvement of the algorithm. Algorithm also show comparisons among the three algorithms and find the best improved Ant-Based algorithm.

The three algorithms are: MDST without local search and without degree constraint, MDST with local search but without degree constraint, MDST with local search and with degree constraint.

In MDST without local search, an edge (i,j) is selected randomly, where j is an adjacent vertex of i[4]. In MDST with local search, all the adjacent edges of vertex i is considered, then the edge with highest pheromone level is selected. In MDST without degree constraint, the edge (i,j) is removed from C and added to tree T if this would create no loop[4]. In MDST with degree constraint, while adding (i,j) edge to tree T another checking is needed. If after adding (i,j), degree of i or degree of j exceeds given parameter k and number of skipped edges(skipped Edge) C is smaller than E-V, the edge, (i , j) is skipped and added to skipped Edge rather than adding it to tree T and each time C is increased by one[4]. This event continues until the entire tree is constructed[4].

7) *Parallel Prim's algorithm*

In this algorithm graph is represented by N×N adjacency matrix. Entry (i , j) contains weight of edge from vertex i to vertex j. If the number of processors is P then this algorithm splits the N×N adjacency matrix into P × N/P sub matrices among all the P processors.

Processor at index 0 is called P_0. P_0 controls the flow of algorithm. The root of minimum spanning tree will always be P_0. Then each processor decides nearest vertex to current minimum spanning tree locally. Each processor informs a single processor about its local nearest vertex. This single processor determines the global nearest vertex to the current minimum spanning tree. This global nearest vertex is winner and each processor will be informed about this winner vertex. Then each processor updates its data structure. This procedure is repeated until all vertices are added in minimum spanning tree.

III. CONCLUSION

Here different techniques for minimum spanning tree are considered. Each technique has different idea to construct minimum spanning tree. In this paper all described techniques works only for undirected graph.

REFERENCES

[1] International *Journal* of Computer and Information Technology, Modified Prim's Algorithm, Sunny Dagar
[2] International Journal of Advanced Research in Computer Science And Software Engineering, A New Efficient Technique to Construct a Minimum Spanning Tree, Ardhendu Mandal, Jayanta Dutta and S.C.Pal
[3] IEEE Congress on Evolutionary Computation, An Effective Ant-Based Algorithm for the Degree-Constrained Minimum Spanning Tree Problem, Minh N. Doan
[4] IOSR Journal of Computer Engineering, An Improved Ant-Based Algorithm for Minimum Degree Spanning Tree Problems, Md.Akkas Ali
[5] Parallel Prim's algorithm on dense graphs with a novel extension, Ekaterina Gonina and Laxmikant V. Kal´e, 2007
[6] http://delab.csd.auth.gr/~manolopo/graph/Lec9_MST.ppt
[7] Introduction to Algorithms by Thomas H. Cormen
[8] *www.eecs.berkeley.edu/~egonina/docs/PrimsAlgINTL07.pdf*
[9] http://en.wikipedia.org/wiki/Minimum_spanning_tree
[10] http://en.wikipedia.org/wiki/Prim's_algorithm