# Spanning Tree- Properties, Algorithms and Applications 

K. Lakshmi ${ }^{1 *}$,T. Meyyappan ${ }^{2}$<br>${ }^{1}{ }^{*}$ Department of MCA, Sir M Visvesvaraya Institute of Technology, Bangalore, India<br>${ }^{2}$ Department of Computer Science and Engineering, Alagappa University, Karaikudi, India<br>*Corresponding Author: lak.kesavan@gmail.com<br>Available online at: www.ijcseonline.org<br>Received: 15/Sep/2017, Revised: 28/Sep/2017, Accepted: 19/Oct/2017, Published: 30/Oct/2017


#### Abstract

$\overline{\text { Abstract- In this paper, we present a survey of the spanning trees. The general properties of spanning trees, algorithms for }}$ generation of all possible spanning trees from a graph and minimum spanning tree algorithms are discussed in this paper. The purpose of this study is to give fundamental details on the spanning trees and related work done based on their application domains. The application domains include computer networks, bio-informatics, image processing etc. It is found that research related to spanning trees can be related to the area of graph mining.


$\underline{\text { Keywords-Graph, Spanning Tree, Minimum Spanning Tree }}$

## I. INTRODUCTION

Graph theory has many applications in the field of computer science. Finding Minimum spanning trees of a graph, Hamiltonian circuit has direct application on many real world optimization problems. Many algorithms exist to find out the set of all possible spanning trees from a given graph. Prim's algorithm and Kruskal's algorithm are the two standard algorithms based on greedy technique to identify the minimum spanning tree from a graph.
It is understood that, there are very few algorithms for generating spanning trees from more than two graphs simultaneously. There is a need for algorithms which can handle more than one graph simultaneously because of the application domains of the spanning trees. In this paper, we present a survey on the spanning trees. The paper is organized as follows. In Section 2, terminologies and notations used properties of the spanning trees, mathematical theorems and formulas related to spanning trees are discussed. A study on the algorithms for generating all possible spanning trees and minimum spanning trees are presented in Section 3. A discussion on the different application domains of spanning trees in presented in Section 4. Conclusion and the possible research directions based on spanning trees are presented in Section 5.

## II. TERMINOLOGY

Graph: Graph is a collection Vertices and edges. $\mathrm{G}=\langle\mathrm{V}, \mathrm{E}\rangle$. Spanning Tree: Spanning trees are special subgraphs of a graph. A spanning tree of a graph $G$ is a subset of $G$, which has all the vertices covered with minimum possible number of edges. So, a spanning tree is a connected, acyclic graph with minimum number of edges.

Complete Graph: A Graph in which there is a path from each vertex to every other vertex is a complete graph.
Connected Graph: A connected graph is simply a graph that necessarily has a number of edges that is less than or equal to the number of edges in a complete graph with the same number of vertices.
Tree: A connected graph is simply a graph that necessarily has a number of edges that is less than or equal to the number of edges in a complete graph with the same number of vertices.
Minimum spanning tree: A minimum spanning tree for an unweighted graph $G$ is a spanning tree that minimizes the number of edges. A minimum spanning tree for a weighted graph $G$ is a spanning tree that minimizes the edge weights.

## III. PROPERTIES OF SPANNING TREES

A connected graph G can have more than one spanning tree.
All possible spanning trees of graph $G$ have the same number of edges and vertices.

The spanning tree does not have any cycle (loops).
Spanning tree is minimally connected.
Spanning tree is maximally acyclic.
Spanning tree has $\mathrm{n}-1$ edges, where n is the number of vertices.

From a complete graph, by removing maximum e - $\mathrm{n}+$ 1 edge, we can construct a spanning tree.
A complete graph can have maximum $\left.\mathrm{n}^{(\mathrm{n}-2}\right)$ number of spanning trees.


Fig. 1 Graph G1 and its all possible spanning trees S1, S2, S3, S4, S5, S6

## A. Total number of possible spanning trees from a Graph

## Kirchhoff's matrix tree theorem

The total number of possible spanning trees from a graph is given by Kirchhoff's matrix tree theorem. It is calculated as below.

STEP 1: Create Adjacency Matrix for the given graph.
STEP 2: Replace all the diagonal elements with the degree of nodes. For eg. Element at $(1,1)$ position of adjacency matrix will be replaced by the degree of node 1 , element at $(2,2)$ position of adjacency matrix will be replaced by the degree of node 2, and so on.
STEP 3: Replace all non-diagonal 1's with -1 .
STEP 4: Calculate co-factor for any element.
STEP 5: The cofactor of any element in the matrix is the same and that is the total number of spanning trees for that graph.


Figure 2. Graph G2
Adjacency matrix of the given Graph is


Replacing all the diagonal elements with degree of the corresponding vertex


Replacing all the non-diagonal elements that 1 with -1

| 3 | -1 | -1 | -1 |
| :---: | :---: | :---: | :---: |
| -1 | 2 | 0 | -1 |
| -1 | 0 | 2 | -1 |
| -1 | -1 | -1 | 3 |

Finding the Co-factor of any element in the above matrix

Co-factor of 3 is the determinant of the matrix
20-1
02-1
$-1-13$

$$
\begin{aligned}
& =2(2 \times 3-(-1)(-1))-0(0 \times 3-(-10(-1))-1(0 x-1-2 \times(-1)) \\
& =2(6-1)-0-1(0+2) \\
& =2(5)-0-1(2) \\
& =10-0-2 \\
& =8
\end{aligned}
$$

Hence, the total number of possible spanning trees for the given Graph is 8 .

## Cayley's Formula

The total number of possible spanning trees from a graph can be calculated based on Cayley's formula in Graph theory. It states that the number of different possible labelled trees with n vertices is $\mathrm{n}{ }^{\mathrm{n}-2}$.


Figure 3. A Complete Graph $G$ and its all possible spanning trees $a, b, c$
Number of vertices in the Graph $\mathrm{n}=3$
Total number of possible spanning trees is $n^{n-1}=3^{3-1}$

## IV. SPANNING TREE ALGORITHMS

Spanning tree algorithms are of two categories. Algorithms that generate all possible spanning trees of a Graph and algorithms that generate the minimum spanning tree.

## A. Algorithms for generating all the Spanning trees from a Graph

Many problems in science and engineering can be formulated in terms of graphs. There are problems where spanning trees are necessary to be computed from the given graphs. Connected subgraph with all the $n$ vertices of the graph $G(\mathrm{~V}, \mathrm{E})$, where $|\mathrm{V}|=n$, having exactly of $n(\mathrm{n}-1)$ edges called the spanning tree of the given graph. The major bottleneck of any tree generation algorithm is the prohibitively large cost of testing whether a newly born tree
is twin of a previously generated one and also there is a problem that without checking for circuit , generated subgraph is tree or non-tree. This problem increases the time complexity of the existing algorithms.

In 1975, Read and Tarjan, proposed a backtrack algorithm for listing Spanning Trees. The algorithm proceeds by avoiding generation of subgraphs that may not lead to a spanning tree.

Gabow \& Myers in 1978 proposed an algorithm for finding all spanning trees (arborescences) of a directed graph [9]. It uses backtracking and a method for detecting bridges based on depth- search. The time required is $\mathrm{O}(\mathrm{V}+\mathrm{E}+\mathrm{EN})$ and the space is $\mathrm{O}(\mathrm{V}+\mathrm{E})$, where $\mathrm{V}, \mathrm{E}$, and N represent the number of vertices, edges, and spanning trees, respectively. If the graph is undirected, the time decreases to $\$ \mathrm{O}(\mathrm{V}+\mathrm{E}+$ VN ), which is optimal to within a constant factor.

Enumeration of spanning trees of an undirected graph is one of the graph problems that have received much attention in the literature. In 1986, Winter and Pawel developed an algorithm based on the idea of contractions of the graph [21]. The worst-case time complexity of the algorithm is $\mathrm{O}(\mathrm{n}+\mathrm{m}$ +nt ) where n is the number of vertices, m the number of edges, and the number of spanning trees in the graph. The worst-case space complexity of the algorithm is $\mathrm{O}\left(\mathrm{n}^{2}\right)$.

On the other hand, Matsui in 1993 developed an $\mathrm{O}(\mathrm{N} \mathrm{V}+\mathrm{V}$ $+\mathrm{E})$ time and $\mathrm{O}(\mathrm{V}+\mathrm{E})$ space algorithm for enumerating all spanning trees explicitly, by applying the reverse search scheme [15]. Reverse search is a scheme for general enumeration problems. For outputting all the spanning trees explicitly, this algorithm is optimal.

Shioura \& Tamura in the year 1995 considered the problem of enumerating all spanning trees of G [21]. In order to explicitly output all spanning trees, the output size is of $\mathrm{O}(\mathrm{NV})$, where N is the number of spanning trees. They proposed a new algorithm for representing the output in a compact form. According to this the output size can be compressed into $\mathrm{O}(\mathrm{N})$ size. The algorithm is optimal in the sense of time complexity.

In 1995, Kapoor and Ramesh proposed an algorithm for generating all spanning trees from undirected and weighted graphs [11]. The $\mathrm{O}(\mathrm{N}+\mathrm{V}+\mathrm{E})$ time and $\mathrm{O}(\mathrm{V}$ E) space algorithm adopted the `compact' output, which is optimal in the sense of time complexity.

Matsui and Tomomi in 1998 proposed an algorithm for finding all the spanning trees in undirected graphs [17]. The algorithm requires $\mathrm{O}(\mathrm{n}+\mathrm{m}+$ on) time and $\mathrm{O}(\mathrm{n}+\mathrm{m})$ space, where the given graph has $n$ vertices, $m$ edges and $\varnothing$
spanning trees. This is an optimal algorithm to output all the spanning trees explicitly.

Again, Kapoor and Ramesh, in 2000 presented an $O(N V+$ $V^{3}$ ) time algorithm for enumerating all spanning trees of a directed graph. This improves the previous best known bound of $O(N E+V+E)$ when $V^{2}=O(N)$, which will be true for most graphs. Here, $N$ refers to the number of spanning trees of a graph having $V$ vertices and $E$ edges. The algorithm is based on the technique of obtaining one spanning tree from another by a series of edge swaps.
Burcu Bozkurt and Durmuş Bozkurt in 2014 established some bounds for the number of spanning trees of connected graphs [3]. It was in terms of the number of vertices, the number of edges, maximum vertex degree, minimum vertex degree, first Zagreb index.
An Algorithm to Generate All Spanning Trees of a Connected and Undirected Simple Graph using divide and conquer approach was proposed by Chakraborty, Maumita \& Hazra, Goutam \& Pal, Rajat in 2017 [4].

## B. Algorithms for finding minimum spanning tree of a Graph

There are several greedy algorithms for finding a minimal spanning tree M of a graph. Kruskal's and Prim's algorithms are well known. Kruskal's algorithm proceeds by adding shortest edge one at time, in such a way that it does not form a circuit with edges already in it. In Prim's algorithm, the spanning tree generation begins by choosing a shortest edge. In each step a shortest edge between a vertex in the generated tree and a vertex not in it is added until the tree has n-1 edges.

Although both are greedy algorithms, they are different in the sense that Prim's algorithm grows a tree until it becomes the MST, whereas Kruskal's algorithm grows a forest of trees until this forest reduces to a single tree, the MST. The disadvantage of these algorithms is that they may not give the best solution always.

## V. APPLICATIONS

## A. Network design

Minimum spanning trees are useful in applications like telephone, electrical, hydraulic, TV cable, computer, road network design. One standard application can be a problem like phone network design. If a number of offices are to be connected using leased lines and the phone company charges different amounts of money to connect different pairs of offices depending on the difficulties involved in laying phone lines. Then the telephone network that has to be installed can be a minimum spanning tree.

## B. Cluster Analysis

K clustering problem can be viewed as finding a Minimum spanning tree and deleting the k-1 most expensive edges.

## C. Brain Network Analysis

Analysis of the minimum spanning tree is useful in the comparison of brain networks. The advantage is that, it avoids methodological biases. Even though the minimum spanning tree does not utilize all the connections in the network, it still provides a, mathematically defined and unbiased, sub-network with characteristics that can provide similar information about network topology as conventional graph measures.

## D. Data Storage of Amino acids in Protein Structure:

A clustering method based on Minimum Spanning Treebased was proposed by Karthikeyan et al., in 2012[7]. Firstly, the $\mathrm{N} \times \mathrm{N}$ distance matrix is constructed, where; N is the number of proteins in the dataset. Then the complete graph is constructed from the distance matrix. In the complete graph each node is associated with a single protein to be clustered. The distance of the protein from all the remaining proteins in the dataset is calculated using Euclidian distance method. Minimum Spanning Tree is constructed from the complete graph using Prim's algorithm. The weight of each edge in the Minimum Spanning Tree is the distance between the connected protein nodes. This minimum spanning tree is used for clustering the proteins.

## E. Broadcasting in Computer Networks

In Ethernet network, information is broadcasted to all the nodes in the network using Spanning tree protocol. In large computer networks, it is useful in constructing trees for broadcasting information to all the nodes in the network.

## F. Other Applications

There are many other applications were information on the minimum spanning tree or the set of all possible spanning trees are useful of the input network data set is useful. Image registration and segmentation, Curvilinear feature extraction in computer vision, Handwriting recognition of mathematical expressions, Circuit design: Regionalization of sociogeographic areas, the grouping of areas into homogeneous, contiguous region, Comparing ecotoxicology data, max bottleneck paths, LDPC codes for error correction, learning salient features for real-time face verification, reducing data storage in sequencing amino acids in a protein, model locality of particle interactions in turbulent fluid flows, autoconfig protocol for Ethernet bridging are some examples.

## VI. CONCLUSION AND FUTURE SCOPE

In this paper, we have discussed on the characteristics of a spanning trees. Mathematical theorems and formulas related to spanning trees were discussed. A study on the existing algorithms (i) for generation of all spanning trees (ii) minimum spanning trees is presented. Spanning trees and
also minimum spanning trees are useful in many application domains. Existing algorithms were focused on the generation of spanning trees from a single graph or at most from two input graphs simultaneously. Design of algorithms that can generate all possible spanning trees or minimum spanning trees from a set of input graphs can be taken for further research. In this paper, we have discussed on the characteristics of a spanning trees. Mathematical theorems and formulas related to spanning trees were discussed. A study on the existing algorithms (i) for generation of all spanning trees (ii) minimum spanning trees is presented. Spanning trees and also minimum spanning trees are useful in many application domains. Existing algorithms were focused on the generation of spanning trees from a single graph or at most from two input graphs simultaneously. Design of an algorithm that can generate all possible spanning trees or minimum spanning trees from a set of input graphs can be taken for further research. This work can provide the fundamental information required for extending the research in the area of graph mining or pattern mining to mining of spanning trees.

## References

[1] Alexander K. Kelmans., On graphs with the maximum number of spanning trees 1996.
[2] AR. PonPeriasamy, E. Thenmozhi, A Brief survey of Data Mining Techniques Applied to Agricultural Data, International Journal of Computer Sciences and Engineering, 4(5), 2017.
[3] Burcu Bozkurt and Durmuş Bozkurt, "On the Number of Spanning Trees of Graphs," The Scientific World Journal, Vol. 2014, Article ID 294038, 5 pages, 2014. doi:10.1155/2014/294038
[4] Chakraborty, Maumita \& Hazra, Goutam \& Pal, Rajat, Divide-and-Conquer: An Approach to Generate All Spanning Trees of a Connected and Undirected Simple Graph. 700-91,2017
[5] Chen, W. K., On directed trees and directed $k$-trees of a digraph and their generation, SIAM J. Appl. Math. 14 550-560, 1966
[6] Das et al., The number of spanning trees of a graph, Journal of Inequalities and Applications, 2013.
[7] Dr. T. Karthikeyan, S. John Peter, B. Praburaj (2012) Minimum Spanning Tree-based Clustering Applied to Protein Sequences in Early Cancer Diagnosis, International Journal of Computer Science And Technology, Vol. 3, Issue 1, Jan. - March 2012.
[8] Gabow, H.N., Two algorithms for generating weighted spanning trees in order. SIAM Journal on Computing, 6(1), 139-150., 1977.
[9] Gabow, H.N. \& Myers, E.W., Finding all spanning trees of directed and undirected graphs. SIAM Journal on Computing, 7, 280-287. 1978.
[10] Hakimi, S.L., On trees of a graph and their generation. Journal of the Franklin Institute. 272. 347-359. 10.1016/0016-0032(61)90036-9.,1961
[11] Kapoor, S. \& Ramesh, H., Algorithms for enumerating all spanning trees of undirected and weighted graphs. SIAM Journal on Computing, 24, 247-265.,1995
[12] Kapoor, S. \& Ramesh, H. An algorithm for enumerating all spanning trees of a directed graph. Algorithmica, 27(2), 120130.,1997.
[13] Kocay, William; Kreher, Donald L., "5.8 The matrix-tree
theorem", Graphs, Algorithms, and Optimization, Discrete Mathematics and Its Applications, CRC Press, pp. 111-116, ISBN 978-0-203-48905-5., 2004.
[14] Matsui, An algorithm for finding all the spanning trees in undirected graphs. Technical Report METR 93-08, Dept. of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, 1993.
[15] Matsui, T., A flexible algorithm for generating all the spanning trees in undirected graphs. Algorithmica, 18(4), 530-543., 1997.
[16] Matsui, Tomomi., An Algorithm for Finding All the Spanning Trees in Undirected Graphs., 1998.
[17] McDiarmid, Colin; Johnson, Theodore; Stone, Harold S., "On finding a minimum spanning tree in a network with random weights", Random Structures \& Algorithms, 10 (1-2): 187204, MR 1611522.,1997.
[18] Minty, G.J., A simple algorithm for listing all the trees of a graph. IEEE Transactions on Circuit Theory, CT-12, 120, 1975.
[19] Naskar S., Basuli K., Sarma S.S., Generation of All Spanning Trees in the Limelight. In: Meghanathan N., Chaki N., Nagamalai D. (eds) Advances in Computer Science and Information Technology. Computer Science and Information Technology. CCSIT 2012.
[20] Shioura, A. \& Tamura, A, . Efficiently scanning all spanning trees of an undirected graph. Journal of the Operations Research Society of Japan, 38(3), 331-344., 1995.
[21] S. Joshi , F.U. Khan, N. Thakur., Contrasting and Evaluating Different Clustering Algorithms: A Literature Review.,International Journal of Computer Sciences and Engineering., Volume-2, Issue-4 , Page no. 87-91, Apr-2014.
[22] Winter, Pawel,. An Algorithm for the Enumeration of Spanning Trees.. BIT. 26. 44-62. 10.1007/BF01939361.1985.
[23] Wu, Bang Ye; Chao, Kun-Mao, Spanning Trees and Optimization Problems, CRC Press, ISBN 1-58488-436-3,2004.

## Authors Profile

$K$ Lakshmi completed her M.C.A degree in 1998, from Alagappa University, Karaikdudi, Tamilnadu, India. She is currently pursuing Ph.D. currently working as Associate Professor in the Department of MCA, Sir M.Visvesvaraya Institute of Technology, Bangalore, Karnataka, India. She has published papers in international
 journals presented a research paper in an IEEE conference. Her main research work focuses on Frequent sub graph mining Algorithms. She has 18 years of teaching experience.

Dr.T.Meyyappan, pursued his Ph.d from Alagappa University, Karaikudi, TamilNadu, India.He is currently working as Professor in the Department of Computer Science and Engineering, Alagappa University, Karaikudi, Tamil Nadu, India. He is a member of IEEE \& IEEE computer society.His main research work focuses on Algorithms, Network Security, Data Mining and Operation Research. He is the
 recipient of the Best Citizens of India Award 2012, International Publishing House, New Delhi. He has published around 48 research papers in reputed international journals including Thomson Reuters (SCI \& Web of Science) and conferences including IEEE. He has more than 25 years of experience in teaching.

