# A Study on 'Number of Spanning Trees'

Adarsh Kumar Verma Student,Galgotias College Of Engineering and Technology Greater Noida, G. B. Nagar 201306, India Saurabh Sharma Assistant Professor, Galgotias College Of Engineering and Technology Greater Noida, G. B. Nagar 201306, India Anuj Tiwari Student, Galgotias College Of Engineering and Technology Greater Noida, G. B. Nagar 201306, India

# ABSTRACT

There exist many algorithms for producing the spanning trees of a graph with better time and space complexities. In this research study, we are presenting a study on number of spanning trees and a technique based on the basic cycle to find the number of spanning trees and also the structure of all the spanning trees of a labeled and undirected graph.

# **General Terms**

Self loop, Parallel edges, Cutset.

## Keywords

Basic cycle, Internal edges, External edges.

## 1. INTRODUCTION

Many problems and their applications in mathematics and networking are based on the graph and the spanning trees of the graph. A tree T is said to be a spanning tree of a connected, undirected graph G with n vertices if T is a sub graph of G and it have exactly (n-1) edges of G. A graph G can have many spanning trees. Computation of all possible spanning trees of a graph is considered to be a classical problem in graph theory. Many researchers have shown their interest in this problem and had also come up with new ideas and algorithms. There are many research techniques adopted for the evaluation of all the spanning trees of a graph, can be broadly classified into – tree testing method [1, 7], elementary tree transformation [2, 3, 4, 5, 8], and successive reduction method [4, 9] of graphs. But we have proposed a simple approach, we will divide the graph in several sub graphs and then evaluate spanning trees for each of them.

# 2. BACKGROUND

## 2.1 Graph

A graph G=(V, E) consists of set of objects vertices  $V=\{v1, v2, v3...vn\}$  and set of edges  $E = \{e1, e2, e3...en\}$ . In the graph there are vertices and edges, vertices are connected by edges with each other. The edges may be directed or undirected, weighted or non-weighted. For example, the cities like Delhi, Lucknow, Kolkata, Mumbai are supposed to be the nodes and they are connected by roads can be supposed as edges, and the set of cities and roads is graph.

## 2.2 Spanning trees

A tree T is said to be a spanning tree of a connected graph G having n vertices and e edges if T is a sub graph of G and having all vertices of G and n-1 edges of graph G. The algorithms of finding minimum spanning tree of a graph are widely used in networks but here we will discuss number of

spanning trees of a graph. If we have solution to the problem that how many number of spanning trees of a graph will be then we have options to choose the particular path from one node to another node in the network.

# 2.3 Self Loop

Self loop can be called as edges in a graph G whose starting vertex and end vertex is same. A simple graph has no self loop. If a graph has self loop then the graph will be called as multi-graph. The role of self loop in the spanning trees is that if there is any self loop present in the graph then we will discard that loop for each spanning tree because self loop will make a cycle.

# 2.4 Parallel Edges

If a graph G=(V, E) consists of two or more edges, that are incident on the same vertices are called as parallel edges. Parallel edges are like the two or more paths between two cities. Significance of mentioning parallel edges is that for each parallel edge there will be same number of spanning tress.

## 2.5 Basic Cycle

A basic cycle is the cycle in a graph G which contains no other cycle in it. There is much significance of the basic cycle in our study. Fig. 1 represents the basic cycles.



**1** Basic Cycle

3 Basic Cycles

Fig. 1 : Examples of Basic Cycles

## 2.6 Internal Edges and External Edges

In Fig. 2 internal and external edges are mentioned, the edges which define the polygon are external edges and the edges, which are inside the polygon and connect the nodes, are internal edges. The external edges are ab, bc, cd, de, ef, fa and internal edges are ac, ce.





Fig. 4 : Structures of Graph

## **3.2 Basic Cycles Identification**

For each structure we will identify the basic cycles, because the basic cycles play an important role in finding the spanning trees of the graph. An area surrounded by minimum 3 edges and having no cycle within it is basic cycle. Now, evaluating basic cycles, for Structure 1, having only one basic cycle which is surrounded by 5 edges abcdea, Structure 2 have 2 basic cycles abca and acdea. Structure 3 have 3 basic cycles abca, acda, adea. Structure 4 have 5 basic cycles abca, acda, adea, baeb, bcdeb.



Structure 1:1 Basic Cycle Structure 2:2 Basic Cycles



Structure 3:3 Basic Cycles Structure 4:5 Basic Cycles



Fig. 2 : Example of internal and external edges

## 3. PROPOSED METHOD

Number of spanning trees and their structure is evaluated in three steps :

## 3.1 Creation of Structures

Since our work is related to complete graph, we have created a base line in the form of complete graph. Breaking the graph into its sub graphs on the basis of internal edges and these sub graphs will be called as structures of the graph. Structures are created to just divide the problem in several sub problems. We will take the combination of the internal edges to create the structures. Initially we will take none of the internal edges, then one, then two and so no. Fig. 3 shows a graph having 5 vertices and 8 edges. Now dividing the edges into two groups, internal edges and external edges. In the graph internal edges are ac, ad, be and external edges are ab, bc, cd, de, ea.



Structure are evaluated including internal edges one by one, from the fig. 4 structure 1 having 0 internal edges, structure 2 has 1 internal edges, structure 3 has 2 internal edges and structure 4 has all three internal edges. If there are n internal edges then total number of structures can be calculated as

Total number of structures of graph =  ${}^{n}C_{0} + {}^{n}C_{1} + {}^{n}C_{2} + \ldots + {}^{n}C_{n}$ 

Here there are 3 internal edges we will have 1 structure with 0 internal edges, 3 structures with 1 internal edge, 3 structures with 2 internal edges and 1 structure with all the 3 internal edges, hence, total number of structures

$$= {}^{3}C_{0} + {}^{3}C_{1} + {}^{3}C_{2} + {}^{3}C_{3} = 1 + 3 + 3 + 1 = 8$$

## 3.3 Evaluating Spanning Trees

For each structure we will evaluate the spanning trees by the combination of the basic cycle and edges which are responsible for the cycle. By the elimination of edges involved in basic cycle we can get the spanning trees.

#### 3.3.1 Structure 1

The spanning trees for the structure can be calculated by the basic cycle present in the structure 1. The basic cycle is made by all the 5 edges, hence by removing 1 edge at a time, breaking of cycle will take place, and spanning tree will form. Since there are 5 edges, we have to remove 1 edge at a time. Hence there will be  ${}^{5}C_{1}$  combination. All the spanning trees evaluated are shown in Fig. 6.



#### Fig. 6 : SPT of Structure 1

#### 3.3.2 Structure 2

The type 2 structure having one internal edge and due to this there are 2 basic cycles. Evaluating spanning trees for structure 1, we will not remove the internal edges because by doing this we will be repeating the spanning trees which are already evaluated in structure 1. From fig. 7 spanning trees of 1 structure of type 2. For the creation of spanning trees we have to remove 2 edges to break the basic cycles. Since there are 3 structures of type having 1 internal edge, and this edge forms 2 basic cycles and first basic cycles there are 2 external edges and in second there are 3 external edges.

Spanning trees for structure 2 = 3\*2 = 6

For all 3 structures of type 2, number of spanning trees =3\*6

= 18

Now we will calculate manually, the fig. 7 shows the spanning trees calculated manually by the combination of edges.





#### 3.3.3 Structure 3

In the structure 3 type, there are 2 internal edges. Fig 8 shows all the structures of type 3, with the 2 internal edges. Structure ii and iii are isomers of each other, hence there will be same number of spanning trees but structure i will have different number of spanning trees.



Fig. 8 : All Structures of Type 3 Structure

We are evaluating the spanning trees of the structure type 3(i). There are 3 basic cycles in the structure, for evaluating spanning trees we have to remove 1 edge from abca at a time, and one edge from aeda at a time, but from the cycle acda we will have to remove cd edge each time otherwise it will make a cycle.

Total number of spanning trees = 2 \* 2



Fig. 9 : All the spanning trees of structure 3 (i)

Since the structure 3 (ii) and 3 (iii) are isomers of each other hence they will have same number of spanning trees. Here we have evaluated spanning trees for the structure 3 (ii). There are total 7 spanning trees formed. The fig. 10 represents all the spanning trees of structure 3 (ii).



Fig. 10 : All the spanning trees of structure 3 (ii)

Total spanning trees formed with structure type 3

$$= 4 + 7 + 7$$
  
= 18

3.3.4 Structure 4

In the structure 4 we will include all the 3 internal edges. The spanning trees formed, will be the combination of the edges ab, bc, ae, ed, but we have to remove cd edge each time otherwise it will create a cycle.





Fig. 11 : All SPTs of structure 4

Now we will calculate total spanning trees calculated by all the structures.

Number of spanning trees of the graph = number of spanning trees by all the structure 1 type + number of spanning trees by all the structures type 2 + number of spanning trees by all the structure 3 type + number of spanning tree by structure 4

= 5 + 18 + 18 + 4

#### 4. CONCLUSION AND FUTURE SCOPE

In this research study we have represented how by the combination of the internal edges, external edges and basic cycles the spanning trees can be generated. The approach is simple and based on dividing the big problem into small problems. By following the above approach we will get number of spanning trees and their structures also. The approach is suitable for the less complex graphs. We will improve the method for the more complex graph and network structures.

#### 5. ACKNOWLEDGMENTS

Our sincere thanks to the Gaurav Yadav (Student, Computer Science and Engineering Department, Galgotias college of Engineering and Technology) for his motivation and support.

International Journal of Computer Applications (0975 – 8887) Volume 73– No.19, July 2013

## 6. REFERENCES

- Char, J.P., Generation of Trees, Two-Trees and Storage of Master Forests, IEEE Transactions on Circuit Theory, Vol. CT-15, pp. 128-138, 1968.
- [2] Hakimi, S.L., On Trees of a Graph and their Generation, Journal of the Franklin Institute, Vol. 272, No. 5, pp. 347-359, 1961.
- [3] Kapoor, S. and H. Ramesh, Algorithms for Enumerating All Spanning Trees of Undirected and Weighted Graphs, SIAM Journal on Computing, Vol. 24, No. 2, 1995.
- [4] Matsui, T., An Algorithm for Finding All the Spanning Trees in Undirected Graphs, Technical Report: METR 93-08, Department of Mathematical Engineering and Information Physics, University of Tokyo, Tokyo, 1993.

- [5] Mayeda, W. and S. Seshu, Generation of Trees without Duplications, IEEE Transactions on Circuit Theory, Vol. CT-12, pp. 181-185, 1965.
- [6] Minty, G.J., A Simple Algorithm for Listing All the Trees of a Graph, IEEE Transactions on Circuit Theory, Vol. CT-12, pp. 120, 1965.
- [7] Sen Sarma, S., A. Rakshit, R.K. Sen, and A.K. Choudhury, An Efficient Tree Generation Algorithm, Journal of the Institution of Electronics and Telecommunications Engineers, Vol. 27, No. 3, pp. 105-109, 1981.
- [8] Shioura, A. and A. Tamura, Efficiently Scanning All Spanning Trees of an Undirected Graph, Research Report: B-270, Department of Information Sciences, Tokyo Institute of Technology, Tokyo, 1993.
- [9] Winter, P., An Algorithm for the Enumeration of Spanning Trees, BIT, Vol. 26, pp. 44-62, 1986.