Combinatorial Algorithm for Finding Spanning Forests

Nuha Abd-uljabbar and Manal N. Yaqoob
Department of Applied Sciences, University of Technology.

Abstract

In this paper we present a new combinatorial algorithm for finding all different spanning forests for a disconnected graph G, depending on the adjacency matrix, the cyclomatic number, the combination between the numbers of edges and the cyclomatic number, and the permutation between the entries of the adjacency matrix to determine the spanning forests.

Keywords: Spanning trees, combinatorial algorithms.

1-Introduction

The enumeration of spanning trees of a finite graph ranges among the classical takes of combinatorial and has been studied by Kirchhof in 1847 in his famous matrix tree theorem [6].

The tree may represent a data structure for a computer file in which information is stored at places corresponding to the vertices of the graph [2].

The first algorithm for finding a spanning tree was developed by Czekanowski in 1909 [5], then by Otakar Borůvka in 1926 [7], many authors interesting to find a new algorithms for spanning tree, or spanning forest, most of them found an algorithm to calculate the maximum or minimum spanning tree for edges weight graph [4], or found the diameter of the edge weight spanning tree [1]. They are depending on their algorithms on the edge weight and the neighbor edges of the chosen one. Others depending on the largest degree of the vertices in the graph to construct the algorithm [8], they are represent the graph in a computer in the form of an array end points, for more information see [7,10]. In this work we give an algorithm that depends on the adjacency matrix of the given graph and the combinatorial techniques.

Let G=(V,E,K) be a finite, simple, undirected, disconnected graph with order N, the number of vertices V(G) denoted by $|V| = N$, the number of edges E(G) is $|E| = M$, and the number of components (each connected graph) K(G) is $|K| = k$. A *forest* F is defined to be a simple graph (graph with no loops or multiple edges), which contains no circuits (closed sequence of vertices and edges), a connected forest is called a *tree*, T_N, if it has N vertices, for example Fig.(1-1) shows a forest with four components, each of them is a tree.

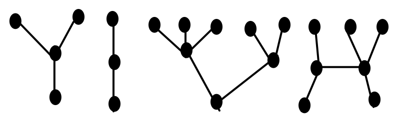

Fig.(1-1) Forest with four trees.

A *subgraph* H of a graph G is simply a graph, all of whose vertices belong to $V(G)$ and all of whose edges belong to $E(G)$.

Spanning subgraph H of a graph G is a subgraph with vertex set V(H)=V(G) and E(H)⊆E(G), thus a spanning subgraph may be specified by its edge set [9].

A spanning subgraph F is a *spanning forest* if each component of F is a connected graph with V vertices and E=V-1 edges which is called a *spanning tree* see Fig.(1-2).

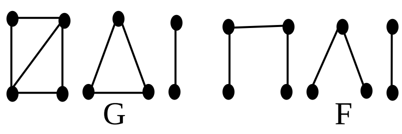

Fig.(1-2) Spanning forest.

A *spanning tree* is defined only for a connected graph because a tree is always connected, and in a disconnected graph of N vertices we cannot find a connected subgraph with N vertices. Each component (which is connected) of a disconnected graph, however, does have a spanning tree. Thus a disconnected graph with k components has a spanning forest with k spanning trees [9].

2-Counting the spanning trees

A tree T_N has exactly N-1 edges which is *bridge* (each edge is essential for the connectedness of the graph).

Thus, if a connected graph G is not a tree, then it has *no bridge*, that is, an edge whose removal does not disconnect the graph. If we iteratively remove no bridge edges until every edge is a bridge of the remaining graph, we get a tree with the same set of vertices as G, and some of its edges which is a spanning tree [1]. The number of the removal no bridge edges equal to (N-M+k) which is called *cyclomatic number*, and denoted by $C_y(G)$ [9].

The numbers of the spanning trees of a connected graph G with N vertices are found by the famous matrix-tree theorem of Kirchhoff which is called Laplacian matrix:

$$L_{ij} = \begin{cases} \deg(v_i) & if \quad i = j \\ -1 & if \quad v_i v_j \in E(G) \\ 0 & otherwise \end{cases}$$

Where deg(v_i) is the degree of the vertex v_i (the number of the incident edges on the vertex v_i).

Let L_i be the reduced Laplacian obtained from L by deleting the ith row and the ith column then $\tau(G) = \det L_i$, where $\tau(G)$ is the number of spanning trees in a graph G [6].

3- Spanning Tree Algorithm

Many authors' interested in an algorithm of spanning tree, and built it in many different ways for its application in many different sciences for more details see [2,8].

In this paper we built this algorithm using the adjacency matrix A(G) [9], Where

$$A(G) = \begin{cases} 1 & if \quad v_i v_j \in E(G) \\ 0 & otherwise \end{cases},$$

and depending on:

1) The algorithm for generating r-combinations of {1,2,…,n} in lexicographic order [2].
2) The cyclic permutation, $\rho = (a_1\ a_2\ \dots\ a_n)$.

The cycles in a collection of cycles are disjoint if there is no element of A that appears in the notations for two different cycles of the collection [3].

Algorithm for Finding Spanning Forests

Let G= (V, E, K) be a disconnected graph of order N, and k components C_i, for all $i = 1,2,\dots,k$, $V(C_i) = n$, and $E(C_i) = m$

1. For $i = 1,2,\dots,k$
   a) Calculate adjacency matrix for C_i .
   b) Calculate $C_y(C_i)$, if $C_y(C_i) = 0$ then C_i is a tree and go to 2.
   c) Calculate the total number of graphs (possible different graphs obtained by deleting $C_y(C_i)$ edges) which is equal to the choice of $C_y(C_i)$ edges from m edges $C(m, C_y(C_i)) = \binom{m}{C_y(C_i)}$
   d) For $j = 1,2,\dots,C(m, C_y(C_i))$
      i] Flip (m-n+1) ones to zeros in upper triangle in adjacency matrix above diagonal.
      ii] Flip (m-n+1) ones to zeros in the lower triangle below diagonal (transpose of the flipped ones in upper triangle).
      iii] If there is any row or column of zeros, then it does not represent a tree.
      iv] Otherwise record this matrix as a valid tree.
   e) End j (Repeat steps i-iv for other combination).
2. End i.
3. Calculate the number of forests which is equal to the multiplication of the number of all different trees in each component.
4. Every forest is composed of one tree in each component.
5. End.

The algorithm would be implemented on each component. Number of trees in a forest is the sum of all trees deducted from components.

The algorithm depends on calculating adjacency matrix for each component. Then calculating the cyclomatic number $C_y(C_i)$ the number of edges that must be eliminated from each component, then take every one that ap-pears in the above diagonal of adjacency matrix (the matrix is symmetrical) and numbered them 1, 2… m and then calculate a combination of $C_y(C_i)$ edges out of m. Change each one to zero according to the combination. Now each choice is a tree if the selected graph satisfies the following conditions:

- No zeros row, or zeros column.
- No cyclic permutation of position of ones.

Notes

- A zeros row or a zeros column represents an isolated vertex (vertex with degree zero).
- A cyclic permutation of position of one's represents a cycle.

The following example explains the previous algorithm:

Let G be the following graph with three components:

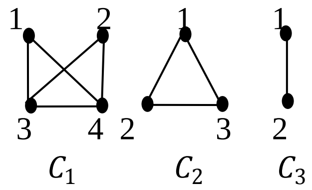

Fig.(3-1) Disconnected graph.

We apply the algorithm on each component:

The first component C_1, which has n=4 and m=5 such that the adjacency matrix of this component is

a) $A = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$,

b) $C_y(C_1) = 5 - 4 + 1 = 2$

c) Compute the combination C(5,2)=$\frac{5!}{2!3!} = 10$, such that the number of graphs we obtained from C_1, when we delete two edges are 10.

d) For the matrix A, if the entry of the upper triangle is one, we labeled these entries by 1, 2, 3, and so on.

Such that the matrix A becomes as follow:

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 3 & 4 \\ 1 & 1 & 0 & 5 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

According to the algorithm for generating 2-combinations of 5 we have:

12	23	34	45
13	24	35	.
14	25	.	.
15	.	.	.

Each choice represent the edges must be delete by changing each one in this position by zero in the adjacency matrix which is explain as follows:

For the first choice 12, replace the ones that labeled by 1 and 2 by zero, so obtained matrix is:

$$A_{12} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix},$$

The obtained graph is

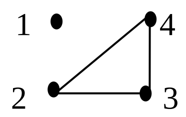

Fig.(3-2) Cyclic graph.

This graph has a cycle then it is not represent a tree and since the matrix has a zeros row we obtain an isolated vertex, such that this case is cancel.

For the second choice 13 , replace the ones that labeled by 1 and 3 respectively by zero, so obtained matrix is:

$$A_{13} = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix},$$

The obtained graph

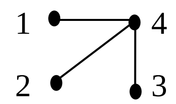

Fig.(3-3) Spanning tree .

Which is satisfies the conditions of the algorithm and represent a tree.

For the choices 14 ,15 ,23 ,24 ,25 ,34, 35 , we obtained six different trees.

But for the choice 45, we obtained a cycle then it is not represent a tree, such that this case is cancel.

Such that we obtain 8 trees for the first component which is equal to $\tau(C_1)$.

For the second component C_2, when we apply the algorithm we obtained three trees which is equal to $\tau(C_2)$.

For the third component C_3, when we apply the algorithm we obtained only one tree which is equal to $\tau(C_3)$.

The number of the forest is $8 \times 3 \times 1 = 24$ as following:

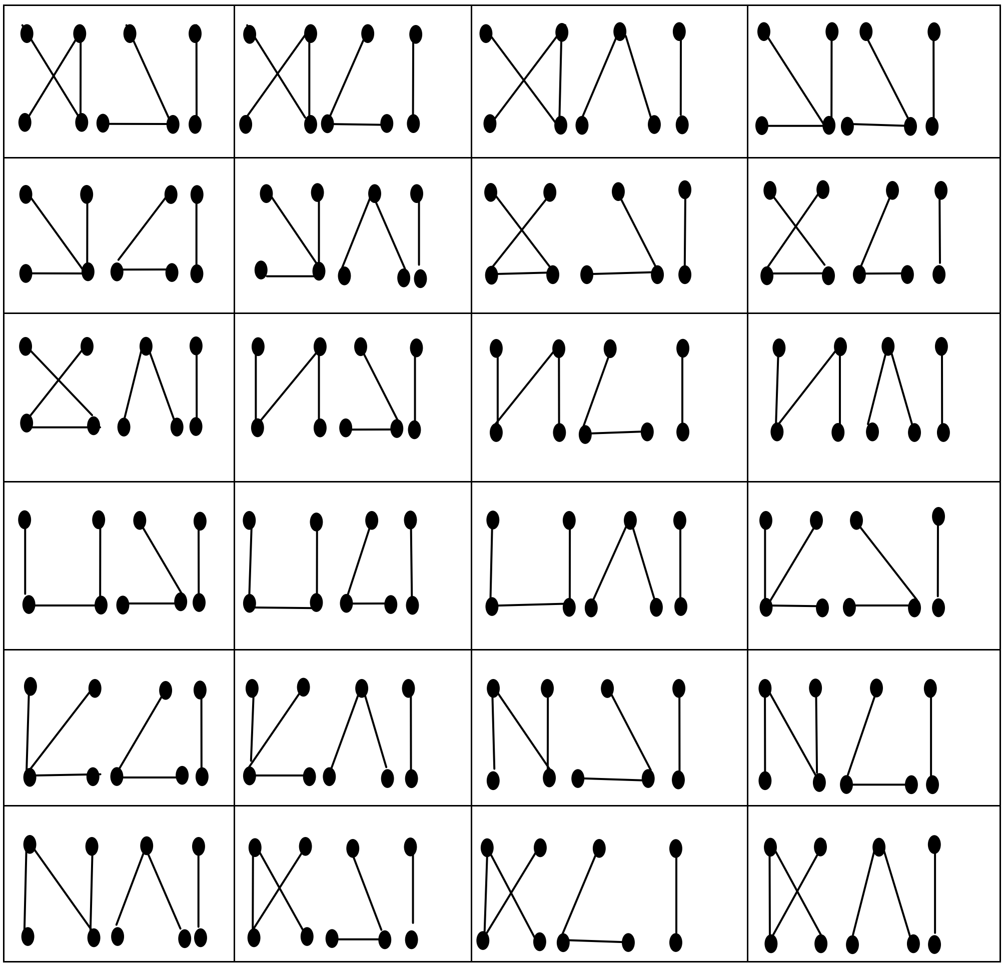

Fig.(3-4) Different spanning forest.

4-Refrence

[1] Addairio-Berry L., Broutin N., and Reed B. "The Diameter of the Minimum Spanning Tree of a Complete Graph" DMTCS proc. AG, 2006, pp. 237-248.

[2] Brualdi R. A., "Introductory Combinatorics", Pearson Education, Inc, 2004.

[3] Fraleigh J. B., "A First Course in Abstract Algebra" third edition, 1982.

[4] Gallager R. G., Humblet P. A., Spira P. M., "A Distributed Algorithm for Minimum-Weight Spanning Trees", ACM Trans. Prog. Lang. Vol.5, No.1, 1983, pp. 66-77.

[5] Greenberg H.J. "Greedy Algorithms for Minimum Spanning Tree" University of Colorado at Denver, http://www.cudenver.edu/~hgreenbe/, March 28, 1998.

[6] Nikolopoulos S.D. and Papadopoulos C. "On the Number of Spanning Trees of $K_n^m \pm G$ Graphs" DMTCS Vol. 8, 2006, pp. 235-248.

[7] Pettie S. and Ramachandran V. "An Optimal Minimum Spanning Tree Algorithm" ACM J., Vol. 49, No. 1, 2002, pp. 16-34.

[8] Rahman M. S. and Kaykobad M., "Independence Number and Degree Bounded Spanning Tree", Appl. Math. E-Notes, Vol.4, No.1,2004, pp.122-124, http://www.math.nthu.edu.tw.

[9] West D. B. "Introduction to Graph Theory", Prentice Hall of India, 2001.

[10] Zsakó L. "Variations for Spanning Tree", AMI, Vol. 33, 2006, pp.151-156.

الخلاصة

في هذا البحث تم تقديم خوارزمية توافيقية جديدة لايجاد كل الغابات المولدة المختلفة لبيان غير متصل G. اعتمادا على مصفوفة التجاور والرقم الدوراني والتوافيق بين عدد الحافات والرقم الدوراني والتباديل بين عناصر مصفوفة التجاور لتحديد الغابات المولدة.