Vertex Degree Preserving Spanning Trees in Digraphs

K. Sangavai
Professor, Department of Mathematics
PSG College of Technology
Email ID: sks.maths@psgtech.ac.in

Abstract:

Let D (V, E) be a digraph with the vertex set V and arc set E and let T be a spanning out tree rooted at v of D. Then T is said to be v-Out Degree Preserving Spanning Out Tree (v-ODPSOT) if ${d_T}^O(v) = {d_G}^O(v)$ and it is said to be v-In Degree Preserving Spanning In Tree (v-IDPSIT) if ${d_T}^I(v) = {d_G}^I(v)$. In this paper we initiate the study of such spanning trees. The existence conditions for those trees are given. The procedure for counting such trees is also presented with example. An algorithm for generating the above trees is given along with illustration. Characterization for strongly connected graph is obtained in terms of the defined tree.

Keywords: *Directed Graph, spanning in trees and out trees, reachable sets, vertex subset degree preserving spanning trees, matrix tree theorem.*

1. Introduction

Digraphs are graphs whose edges are having directions. Digraphs are used to model problems where the direction of flow of some quantity (information, traffic, liquid, electrons and so on) is of importance [8]. When there are limits placed on how much of that quantity can flow through a particular directed edge, we obtain a network. A special type of digraph having no directed cycles, called an activity digraph, has weights on the directed edges indicating the duration of a given activity. These digraphs are used to aid in scheduling individual activities that compose a complex project.

The degree-preserving spanning tree (DPST) problem introduced by Broersma.H et al in [4] is, given a connected graph G, find a spanning tree T of G such that as many vertices of T as possible have the same degree in T as in G. This problem is a graph-theoretical translation of a problem arising in the system-theoretical context of identifiability in networks, a concept which has applications in water distribution networks and electrical networks. It was proved that the DPST problem is NP-complete, even restricted to split graphs or bipartite planar graphs. Linear time approximation algorithms for planar graphs were presented. In [9] Peter Damaschke proved the NP-completeness of DPST problem for bipartite planar degree-5 graphs and for planar degree-3 graphs. Randeep Bhatia et. al studied more about the full degree spanning tree in [10]. But in several networks there will be one or some prevailing nodes whose link with their neighbors should be essentially preserved. So A-DPST has been introduced by Anitha. R and Sangavai. K in [1]. A-Degree Preserving Spanning Tree (A-DPST) where A is a nonempty subset of V of a graph is a spanning tree T in which $\deg_G(v) = \deg_T(v)$ is true for all v $\in$ A and extensively studied in [2]. In [3] the particular case of A, a single vertex set considered. The spanning tree T is called v-degree preserving spanning tree (v-DPST) if $\deg_G(v) = \deg_T(v)$, v $\in$ V. And a graph G is defined as degree preservable graph if each of its spanning trees is v-DPST for some v $\in$ V [3]. Graphs that do not have this property are defined as non-degree preservable.

Directed spanning trees have been studied in many fields. For instance, many problems on road and telephone networks have been formulated as some optimization problems of directed spanning trees. Some of them have complicated objective functions, and we can hardly solve them in efficient time. For those problems, one of the simplest approaches is to use enumerating.

In this paper, the concept of vertex degree preserving spanning is extended to digraphs. Since in digraphs, due to the direction on the edges the vertices are having in degree and out degree. Also, the spanning tree of any digraph is defined as in tree and out tree, according to the direction. Let G = (V, E) be a directed graph with vertex set V and arc set E. An arc is specified by both of its endpoints. One of them is called its head and the other is called its tail. A directed spanning tree of G is a spanning tree in which no two arcs share their tails. Each vertex is the tail of exactly one arc of the directed spanning tree except for a special vertex r. We call r the root of the spanning tree.

2. Vertex Degree Preserving Spanning in digraphs

A strongly connected digraph is a digraph in which it is possible to reach any vertex starting from any other vertex by traversing edges in the direction(s) in which they point. The vertices in a strongly connected digraph therefore must all have of in degree at least 1.A vertex v of a simple digraph is said to be reachable from the vertex u of the same digraph, if there exists a path from u to v. The set of vertices which are reachable from a vertex v is said to be the reachable set of v and is written as R(v).

Spanning trees in digraphs

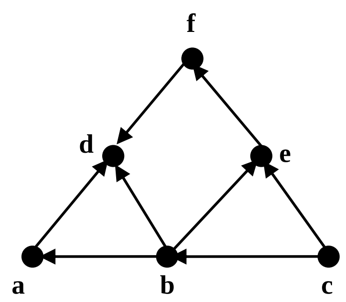
A subgraph T of a connected digraph D is a spanning oriented tree of D if UG (T) is a spanning tree in UG (D), where UG denotes underlying graph.

A sub digraph T of a digraph D is an out-branching or out-tree (in-branching or in tree) if T is a spanning oriented tree of D and T has only one vertex ‘s’ of in-degree (out-degree) zero. The vertex s is the root of T.

v-Out Degree Preserving Spanning Out-Tree (v-ODPSOT) / In-Tree (v-IDPSIT)

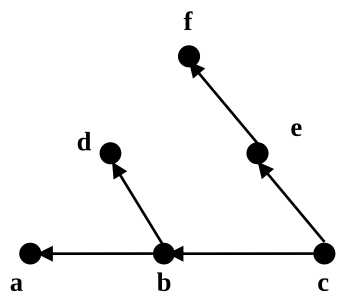
Out-tree rooted at v is said to be a v-Out Degree Preserving Spanning Out Tree (v-ODPSOT) if $\deg_T^O$ (v) = $\deg_G^O$ (v). In a similar way v-In Degree Preserving Spanning In Tree (v-IDPSIT) can also be defined as “In-tree rooted at v is said to be a v-In Degree Preserving Spanning In Tree (v- IDPSIT) if $\deg_T^I$ (v) = $\deg_G^I$ (v)”

Example: A digraph and its c-ODPSOT and d-IDPSIT are given below.



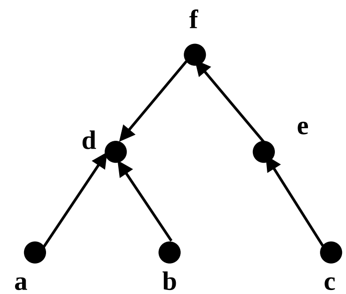
Digraph D

Fig 2.1(a)



c-ODPSOT

Fig 2.1(b)



d-IDPSIT

Fig 2.1(c)

In this section the existence conditions for v-ODPSOT, v-IDPSIT are presented. Also the algorithms of generating such spanning trees are presented and their time complexities are also discussed. A procedure for counting v-ODPSOT is given along with illustration. Other than this a characterization for strongly connected digraph in terms of v-ODPSOT is also given

Theorem 2.1: Existence condition for v-ODPSOT

In any digraph D, v-ODPSOT rooted at v exist if and only if R (v) = V, where V is the vertex set of D.

Proof:

If D has a v-ODPSOT T rooted at v, then T is a connected, circuit less sub graph of D which includes all vertices of D and the out degree of v in T is same as in D.

This means that there exists a directed path between v and other vertices. From definition of spanning out- tree rooted at v, in degree of v is zero.

Therefore, there exist paths from v to all other vertices which would imply that R (v) = V.

Conversely, let R (v) = V

This means that there exists a directed path from v to all other vertices.

Now, the BFS tree rooted at v will be a v-ODPSOT

Theorem 2.2: Existence condition for v-IDPSIT

In any digraph D v-IDPSIT exists if and only if $v \in R(u)$, for all $u \in V\text{-}v$

Proof:

Suppose v-IDPSIT exists and let it be T. By definition T is a spanning tree in which all the vertices are of out degree and the in degree is the same as in D. This implies that there exists a directed path between every vertex belongs to V-v and v. Hence reachable set of u for all u in V-v consists of the element v. (ie) $v \in R(u)$, for all $u \in V\text{-}v$

Conversely, suppose $v \in R(u)$, for all $u \in V-v$

To prove v-IDPSIT exists

$v \in R(u)$, for all $u \in V-v$ implies there exists a directed path from all all $u \in V-v$ to v. Now the BFS in tree rooted at v will be a spanning tree in which the in degree of v is the same as in D.

The following theorem gives a characterization for a strongly connected graph in terms of ODPSOT

Theorem 2.3:

D is strongly connected if and only if v-ODPSOT exists for all $v \in V$, the vertex set of D

Proof:

Assume D as a strongly connected graph. By definition, every vertex in D is reachable from every other vertex in it. In turn implies, reachable set R(v) = V for all $v \in V$

Hence by existence theorem, v-ODPSOT exists for all $v \in V$

Conversely, suppose v-ODPSOT exists for all $v \in V$. Existence theorem implies R (v) = V, for all $v \in V$. That is every vertex of D is reachable from every other vertex. Hence D becomes a strongly connected graph

Theorem 2.4:

In every acyclic digraph D, v-ODPST exists only when it has at most one vertex of in degree zero.

Proof:

In every acyclic graph there must exists at least one vertex with in degree zero.

For that vertex v v-ODPSOT will exists, because that will be the BFS tree rooted at v. If there is one more vertex of in degree zero, then it could not be reachable from v and therefore v-ODPSOT will not exist

Remark:

By similar argument one can prove that in every acyclic digraph v-IDPSIT exists if it has at most one vertex of in degree zero.

Theorem 2.5

v-ODPSOT exists if and only if v is an element of the strong component whose reachable set S is the set of all nodes in SC (D)

Proof:

Let $C_1, C_2, ..., C_k$ are the k-distinct strong components of D such that $C_1 \cup C_2 \cup ... \cup C_k = D$

Let v-ODPSOT exists and v is a vertex belongs to C_1.

To prove R(C_1) = $\{C_2, C_3, ..., C_k\}$, as a contrary, let us consider there exists $C_i \in \{C_2, C_3, ..., C_k\}$ but $C_i \notin$ R(C_1)

v-ODPSOT exists imply by Theorem 2.2 R(v) = V-the set of all vertices. $C_i \notin R(C_1)$ imply the vertices in C_i are not reachable from any of the vertices belongs to $C_1$and hence from v, which is a contradiction

Conversely, suppose C_1 be a strong component whose reachable set $\{C_2, C_3, ..., C_k\}$ and let $v \in C_1$.

To prove: v-ODPSOT exists

Since $R(C_1) = \{C_2, C_3, ..., C_k\}$, all vertices in the other components $C_2, C_3, ..., C_k$ are reachable from every vertex in C_1 and hence from v. Therefore, again by Theorem 2.2 v-ODPSOT exists.

Remark:

Using the above result, just by having the SC(D) one can find the set of all vertices for which ODPSOT exist in D.

Algorithm: 2.6 Constructing v-ODPSOT

1. Let S = { $v \in V$}, v is the root

   T = Φ , initially there are no edges in the tree
2. C = {$N_O(v)$}, C is the current set of vertices being processed
3. l(v) = 0, label the root as 0

   p(v) = v;

   b^* = v;

   Remove b^* from all adjacency lists

   i = 1, initialize variable i used to help label the vertices.
4. For each $w \in N_O(b^*)$, place w in S and place edge (b^*,w) in T.

   Assign successive labels l(w) = i and p(w) = $p(b^*)$, and add w to vertices of C.
5. Define a new b^* to be the vertex x in C such that l(x) is minimum.

   Remove b^* from C and return to step 4.

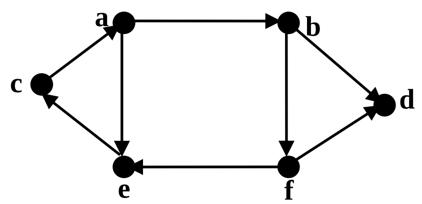
   If however, C is empty stop.

   If every vertex of G has been labeled, a spanning tree has been found.

   If not, then $R(v) \neq V$ and hence no such v-ODPSOT exists.

   Anyway v-ODP maximum component in the digraph D has been found.

Illustration:



Digraph D

Fig. 2.2

Vertex v	**Adjacency List $N_O(v)$**
a	**b,e**
b	**d,f**
c	**a**
d	**-**
e	**c**
f	**d,e**

Table 2.1

1. S = {a}and choose as the root, T = Φ
2. C = { $N_O(v)$ = { b, e }
3. l(a) = 0, l(b) = 1
   $p_o(a)$ = a;
   Let b^* = a, i = 1
   a is removed from all adjacency list

Steps 4 and 5 are explained in the following table:

Pass	**Step**	**b^***	**$x \in S$**	**$e \in T$**	**$y \in C$**	**i**	**New Labels**

1	4
	
	5
		a
		
		b
			a, b, e
			
			f
				ab, ae	b, e
					
					e
						3	l(b) = 1;
							l(e) = 3;
							p(b) = ab;
							p(e) = ae.

2	4
	
	5
		b
		
		e
			a, b, e, d, f	bd, bf	e	4	l(d) = 3;
							p(d) = abd.
							l(f) = 4;
							p(f) = abf

3	4	e	a, b, c, d, e, f	ec	c	5	l(c) = 6; p(e) = aec.

Table 2.2

Now S = V and hence T is arrived. The resultant a-ODPSOT is the following one:

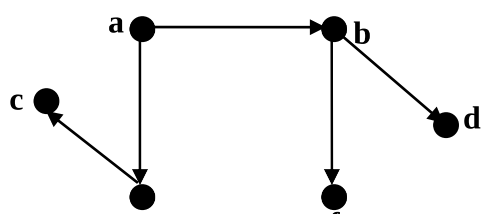


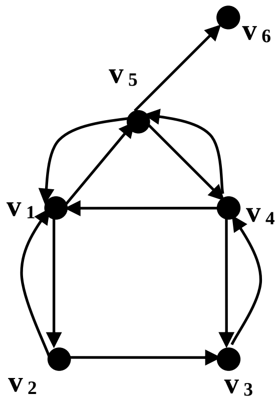
Fig 2.3

Theorem 2.7

Let D be a connected digraph without loops and parallel edges in the same direction. The number of v_i - ODPSOT rooted at v_i is equal to det (N), where N is obtained by subtracting the adjacency matrix A^+ from the in degree matrix D^- and by removing the i^{th} row, i^{th} column and the rows corresponding to the vertices contributing out degree to v_i.

Proof is similar to the proof of the famous matrix tree theorem

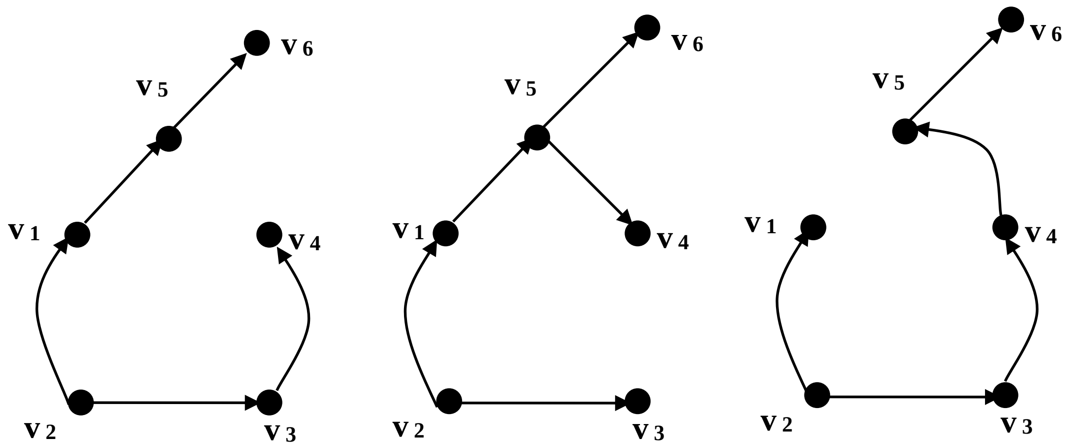
To illustrate the above theorem, consider the following digraph.



Digraph D
Fig 2.4

$$D^- - A^+ = \begin{bmatrix} 3 & -1 & 0 & 0 & -1 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 & -1 & 0 \\ -1 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For example, to get the number of v_2 degree preserving spanning trees, we have delete the 2^{nd} row and second column as well as the rows and columns corresponding to the vertices which contribute out degree to v_2. They are v_1 and v_3. Then the matrix N becomes, $N = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & 0 & 1 \end{bmatrix}$. Now det (N) = 3 and the three v_2-ODPSOT are given below in **Fig 2.5**



Similar to the above result, the number of v_i – IDPSIT can be determined using the following theorem

Theorem 2.8

Let D be a connected digraph without loops and parallel edges in the same direction. The number of v_i - IDPSIT rooted at v_i is equal to det (N), where N is obtained by subtracting the adjacency matrix A^- from the in degree matrix D+ and by removing the i^{th} row, i^{th} column and the rows corresponding to the vertices contributing in degree to v_i.

3.Conclusion

The concept of outdegree and indegree preserving spanning trees can be generalized for any subset of the vertex set of a digraph. The time complexity of the algorithm for generating all v-ODPSOT could be analyzed. Applications for such trees in flow networks could be identified. The other properties like transient closure; for digraphs could be related with the defined spanning trees. The digraphs in which every outree (intree) is v-ODPSOT (IDPSIT) can be defined as out(in)degree preservable digraphs and could be analyzed.

References

[1] Anitha. R and Sangavai. K, “Some Results on Vertex Subset Degree Preserving Spanning Trees”, Proceedings of NCMCM, Department of Mathematics and Computer Applications, PSG College of Technology, pp 466-476, 2005.

[2] Anitha. R and Sangavai. K, “On Vertex Subset Degree Preserving Spanning Trees”, International Journal of Computational and Applied Mathematics, Vol. 2, No. 2, pp 115-123, 2007.

[3] Anitha. R and Sangavai. K, “Degree Preservable Graphs”, International Journal of Mathematics and Computation, Vol. 1; No. N08, pp 36-42, 2008

[4] Broersma.H, Huck.A, Kloks.T, Koppius.O, Kratsch.D, Muller.H, Tuinstra.H, “Degree-preserving spanning forests”, Mathematical Foundations of Computer Science, 1998.

[5] Fred Buckley and Marty Lewinter, “ A friendly introduction to Graph Theory, Pearson Education, Inc, 2003.

[6] Geir Angarsson, Raymond Greenlaw, “Graph Theory: Modelling, Applications and Algorithms”, Pearson Education, Inc, 2007.

[7] Jonathan Gross and Jay Yellen, “Graph Theory and Its Applications”, CRC Press, New York, 1999.

[8] Jorgen Bang- Jensen and Gregory Gutin, “Digraphs: Theory Algorithms and Applications”, Springer-Verlag, 2001.

[9] Peter Damaschke, “Degree-preserving spanning trees in small degree graphs”, Discrete Mathematics 222, pp 51-60, 2000.

[10] Randeep Bhatia, Samir Khuller, Robert Pless, Yoram J. Sussmann, “The Full Degree Spanning Tree Problem – Extended abstract”, Computer Science Department, Technical Report CS-TR-3931, Univ. of Maryland, 1998.