Applied Mathematics E-Notes, 4(2004), 122-124 © ISSN 1607-2510
Available free at mirror sites of http://www.math.nthu.edu.tw/∼amen/

Independence Number And Degree Bounded Spanning Tree[*]

Mohammad Sohel Rahman[†], Mohammad Kaykobad[‡]

Received 3 November 2003

Abstract

This paper deals with the classical degree-bounded-spanning-tree problem. We define degree-d-bounded-spanning tree to be a spanning tree having no vertices of degree larger than d. The main result of this paper states that, if $G = (V, E)$ is a connected graph having independence number d, where $d \geq 2$, then G has a degree-d-bounded spanning tree. Also an $O(n^2)$ algorithm for constructing such a spanning tree is presented.

1 Introduction

A spanning subgraph of a graph $G = (V, E)$ is a subgraph with vertex set V. A spanning tree is a spanning subgraph that is a tree. Spanning trees have been found to be structures of paramount importance in both theoretical and practical problems. As a result spanning trees of a connected graph have been the focus for extensive attention in graph theoretic research. There are many interesting problems on spanning trees. In this paper we are particularly interested in Degree Bounded Spanning Tree problem which is formally defined as follows.

Degree Bounded Spanning Tree Problem: Given a graph $G = (V, E)$ and a positive integer $K \leq |V|$, is there a spanning tree for G in which no vertex has degree larger than K?

The above problem is proved to be NP-Complete [1] and it remains NP-Complete for any fixed $K \geq 2$. NP-Completeness of the above problem suggests that there exists no good characterization for a graph having a degree-bounded-spanning-tree as defined in the problem. A characterization is said to be good if it can be tested easily; polynomially, to be specific. We define the term degree-d-bounded-spanning-tree to be a spanning tree in which no vertex has degree larger than d. In this paper we relate the existence of a degree-bounded-spanning-tree in a graph G to the independence number of G. The main result of our paper states that a graph with independence number d must have a degree-d-bounded spanning tree. In the sequel, we outline a polynomial

[*] Mathematics Subject Classifications: 05C05, 05C85, 68R10.

[†] Department of CSE, BUET, Dhaka-1000, Bangladesh

[‡] Department of CSE, BUET, Dhaka-1000, Bangladesh

time algorithm to construct a degree-d-bounded spanning tree from a graph having independence number d.

Given a graph $G = (V, E)$ and a set $S \subseteq V$, by $G[S]$ we mean the induced subgraph of the set S. $V(G)$ and $E(G)$ are used respectively to denote the vertex set and edge set of G. By $\Pi(G)$ we denote the set of leaf nodes of the graph G. A set $S \subseteq V(G)$ of a graph G is said to an independent set if for every pair of vertices u, $v \in S$, $(u, v) \notin E(G)$. Independence number of G, denoted by $\alpha(G)$, is the cardinality of a maximum Independent set of G. An x, y-path is a path having x and y as the two end vertices of the path. A set of leaves L in a tree T is said to be *initiated* by a nonleaf node v if and only if for every pair x, $y \in L$, v lies in the x, y-path in T. Any notation or definition not explained here can be found in [2].

2 The Main Result

The main result of our paper is the following theorem.

THEOREM 1. Let $G = (V, E)$ be a connected graph such that $\alpha(G)= d$ where $d \geq 2$. Then G has a degree-d-bounded spanning tree.

Before proving Theorem 1, we first present the following important but simple observations.

OBSERVATION 1. Let T be a tree. If there is a $v \in V(T)$ such that $\deg_T(v) = k$, where $k \geq 2$, then there is a set L of leaves in T, such that $|L| = k$ and L is initiated by v.

From Observation 1 we easily get the following observation.

OBSERVATION 2. Let T be a tree. If there is a $v \in V(T)$ such that $\deg_T(v) = k$, then $|\Pi(T)| \geq k$.

Now we are ready to prove Theorem 1. The proof employs construction and induction simultaneously.

PROOF OF THEOREM 1. We first construct an arbitrary spanning tree T_0. Now if for all $v \in V(T_0) = V(G)$, $\deg_{T_0}(v) \leq d$ then we are done. So assume otherwise. Let D be the set of vertices in a spanning tree having degree greater than d. Then, $D(T_0) = \{x | \deg_{T_0}(x) > d\}$. Now we show that there exists a spanning tree T such that $D(T) = \emptyset$. Our strategy is to show it by induction on the number of vertices belonging to D in a spanning tree . We start with a vertex $w \in D(T_0)$. Let in T_0, $\deg_{T_0}(w) = k$, where $k > d$. By Observation 1, there is a set L of leaves of T_0 such that L is initiated by w and $|L| = k$. Since $\alpha(G) = d < k$, there is a pair z, $r \in L$ such that $(z, r) \in E(G)$, because, otherwise L would be an independent set of G with cardinality $k > d$ leading to a contradiction. Since z, $r \in L$ and L is initiated by w, in $T_0 + (z, r)$ there is a cycle C such that $w \in V(C)$. Now we delete an edge (w, p) to break the cycle so as to get a spanning tree T_1. Note that $\deg_{T_1}(w) = \deg_{T_0}(w) - 1$. It is clear that we can apply above procedure until we get a spanning tree T_{k-d} such that $\deg_{T_{k-d}}(w) = d$ after a finite iteration, since $\alpha(G) = d$. So we get a spanning tree T_{k-d} such that $|D(T_{k-d})|$ is one less then $|D(T_0)|$ and this completes our proof.

3 An Efficient Algorithm

In this section we outline a simple algorithm for constructing a degree-d-bounded spanning tree from an input graph G such that $\alpha(G) = d$. We also present a simple worst case analysis of the algorithm.

ALGORITHM 1.
//Input: A connected graph $G = (V, E)$ with $\alpha(G) = d$.
//Output: A degree-d-bounded-spanning tree T.

```
Begin
1. Construct and arbitrary spanning tree $T$ of $G$.
2. if for all $v \in V$, $\deg_T(v) \leq d$, then
3.     return $T$
4. else
5.     Form the set $D$ for $T$.
6.     for each $x \in D$ do
7.           Form the set $L$
8.           while $\deg_T(x) > d$ do
9.                   Find a pair $z$, $r \in L$ such that $(z, r) \in E(G)$
10.                  (Let $C$ is the cycle created in $T+$ $(z, r)$ and $(x, p) \in E(C)$)
11.                  $T = T + (z, r) - (x, p)$
12.          end while
13.    end for
14. end if
15. return $T$
End
```

ANALYSIS: Let $|V| = n$ and $|E| = m$. Constructing an arbitrary spanning tree takes $O(m)$ computational effort. Forming the set D at most takes $O(n)$. To find out how many operations are actually done in the worst case in the for loop of line #6 to line #13, we just need to realize that there can be $O(n^2)$ pairs of z, r to check in line #9 in the worst case. So the overall running time is $O(n^2)$. Thus we get the following theorem.

THEOREM 2. Given a connected graph $G = (V, E)$ with $\alpha(G) = d$, we can find a degree-d-bounded spanning tree in $O(n^2)$ computational effort, where $n = |V(G)|$.

References

[1] M. R. Garey and D. S. Johnson, Computers and Intractability, Freeman, New York, 1979.

[2] D. B. West, Introduction to Graph Theory, Prentice-Hall, Inc., New Jersey, 1996.