Available online at www.sciencedirect.com

SCIENCE DIRECT®

Information Processing Letters 94 (2005) 93–97

Information Processing Letters

www.elsevier.com/locate/ipl

Complexities of some interesting problems on spanning trees

M. Sohel Rahman *, M. Kaykobad

Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh

Received 3 December 2002; received in revised form 18 August 2004

Available online 1 February 2005

Communicated by F.B. Schneider

Abstract

This paper deals with the complexity issues of some new interesting spanning tree problems. Here we define some new spanning tree problems by imposing various constraints and restrictions on graph parameters and present relevant results. Also we introduce a new notion of "set version" of some decision problems having integer $K < |V|$ as a parameter in the input instance, where we replace K by a set $X \subseteq |V|$. For example, the set version of Maximum Leaf Spanning Tree problem asks whether there exists a spanning tree in G that contains X as a subset of the leaf set. We raise the issue of whether the set versions of NP-complete problems are as hard as the original problems and prove that although in some cases the set versions are easier to solve, this is not necessarily true in general.

Keywords: Computational complexity; Decision problem; Graphs; NP-completeness; Spanning trees; Combinatorial problems

1. Introduction

Spanning trees of a connected graph have always been the focus of extensive research. Spanning trees with various constraints and restricted conditions seem to pose various interesting problems. For example, consider the following problems and the known results.

Problem 1.1 (*Degree Constrained Spanning Tree problem*). Given a connected graph $G = (V, E)$ and a positive integer $K < |V|$, we are asked the question whether there is a spanning tree T of G such that no vertex in T has degree larger than K.

Theorem 1.2 (see [2]). *Degree Constrained Spanning Tree problem is* NP-*complete.*

Remark (see [2]). Problem 1.1 remains NP-complete for any fixed $K \geqslant 2$.

Problem 1.3 (*Maximum Leaf Spanning Tree problem*). Given a connected graph $G = (V, E)$ and a positive

* Corresponding author.
E-mail addresses: msrahman@cse.buet.ac.bd (M.S. Rahman), kaykobad_lab@yahoo.com (M. Kaykobad).

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2004.12.016

integer $K < |V|$, we are asked the question whether there is a spanning tree T of G such that K or more vertices in T have degree 1.

Theorem 1.4 (see [2]). *Maximum Leaf Spanning Tree problem is* NP-*complete.*

Remark (see [2]). Problem 1.2 remains NP-complete if G is regular of degree 4 or if G is planar with no degree exceeding 4.

Spanning trees with restrictions on the number of leaves have applications in communication networks and circuit layouts (see [1,4]). In this paper we first introduce some new problems (with relevant new results) where we impose various constraints and restrictions on parameters of spanning trees. Then we introduce a new notion of "set version" of some decision problems having integer $K < |V|$ as a parameter in the input instance, where we replace K by a set $X \subseteq |V|$. For example, the set version of Maximum Leaf Spanning Tree problem asks whether there exists a spanning tree in G that contains X as a subset of the leaf set. The complexities of the set versions of various problems are discussed and we show that remarkably "set versions" of some NP-complete problems are solvable in polynomial times although this may not necessarily be the case all the time.

We denote by $N_G(x)$, the set of vertices (or nodes) which are adjacent to x in graph G and its cardinality by $\deg_G(x)$. The subgraph of G induced by a vertex subset S is denoted by $\langle S \rangle$ and the union of $N_G(x)$ for all $x \in S$ is simply denoted by $N_G(S)$. $V(G)$ and $E(G)$ are sometimes used to denote, respectively, the vertex and edge sets of graph G. For a graph G, we define $\Pi_G = \{v \mid v \text{ is a leaf in } G\}$. The term X-nodes indicates the vertices belonging to the vertex subset X of a graph G. By a matching M of a graph $G = (V, E)$ from A to B, where $A, B \subseteq V$, we mean one such that $M \subseteq E$, every edge of M joins a vertex of A to a vertex of B, every vertex of B is incident with at most one edge of M and every vertex of A is incident with exactly one edge of M. The vertices belonging to the edges of a matching M are said to be saturated by M; the others are unsaturated. A vertex set S is said to be saturated by M if x is saturated by M for all $x \in S$.

2. New problems and results

In this section we introduce some new interesting spanning tree problems and relevant results.

Problem 2.1 (*Minimum Leaf Spanning Tree*). Given a connected graph $G \equiv (V, E)$ and a positive integer $K < |V|$, we are asked the question whether there is a spanning tree T of G such that K or less vertices have degree 1.

Theorem 2.2. *Minimum Leaf Spanning Tree problem is* NP-*complete.*

Proof. This can be easily proved by restricting the value of K to 2, since then we in effect, have to find out a Hamiltonian path in G, which is an NP-complete problem for general graphs [2]. □

Problem 2.3 (*Restricted-Leaf-in-Subgraph Spanning Tree problem*). Given $G \equiv (V, E)$ be a connected graph, X a vertex subset of G and a positive integer $K < |X|$, we are asked the question whether there is a spanning tree T_G such that number of leaves in T_G belonging to X is less than or equal to K.

Theorem 2.4. *Restricted-Leaf-in-Subgraph Spanning Tree* (*RLSST*) *problem is* NP-*complete.*

Proof. We prove this theorem by reduction. It is easy to see that if we assume $X = V$ then RLSST problem reduces to Minimum Leaf Spanning Tree problem (Problem 2.1). Hence the result follows directly from Theorem 2.2. □

We now consider a variant of Maximum Leaf Spanning Tree problem (Problem 1.3) for bipartite graphs.

Problem 2.5 (*Variant of Maximum Leaf Spanning Tree for bipartite graphs*). Let G be a connected bipartite graph with partite sets X and Y. Given a positive number $K \leqslant |X|$, we are asked the question whether there is a spanning tree T_G in G such that number of leaves in T_G belonging to X is greater than or equal to K.

In the remaining of the section we investigate the necessary and sufficient conditions for the existence of such a spanning tree in a bipartite graph as defined in Problem 2.5. First we present the following theorem.

Theorem 2.6. *Let G be a connected bipartite graph with partite sets X and Y and suppose K is a positive number such that $K \leqslant |X|$. Then there is a spanning tree T in G such that number of leaves in T belonging to X is greater than or equal to K if and only if there is a set $S \subseteq X$ such that $|X \setminus S| \geqslant K$ and $\langle S \cup Y\rangle$ is connected.*

Proof. The proof of the theorem is simple. We first consider the "if part". Suppose in G there is a set S such that $|X \setminus S| \geqslant K$ and $\langle S \cup Y\rangle$ is connected. We can easily find a spanning tree T' for the graph $\langle S\cup Y\rangle$. Now suppose in T' number of leaves belonging to S is K'. It is clear that $0 \leqslant K' \leqslant |S|$. Now since G is a connected bipartite graph, for each $x \in X \setminus S$ we can add an edge (x, y) such that $y \in Y$ to get a spanning tree T'' of G. This would mean that for all $x \in X \setminus S$, $\deg_{T''}(x) = 1$. Now since $|X \setminus S| \geqslant K$ so, T'' would necessarily be our desired spanning tree T. Conversely, suppose G has a spanning tree T such that number of leaves in T belonging to X is greater than equal to K. Now let L be the set of leaves of T belonging to X. Since G is bipartite so in T, $N_G(L) \subseteq Y$. Hence if we delete the set L from T_G we still get a tree. This means that in G, $\langle (X \setminus L) \cup Y\rangle$ is connected. Now since according to the assumption $|L| \geqslant K$ so, in effect we have established the existence of the set S ($\equiv X \setminus L$) in G as defined in the conditions. Hence the result follows. □

Now we present a more stringent condition for the existence of desired spanning tree in a bipartite graph G (as defined in Problem 2.5). The conditions are presented in the form of following theorem.

Theorem 2.7. *Let G be a connected bipartite graph with partite sets X and Y and suppose K is a positive number such that $K \leqslant |X|$. Then there is a spanning tree T in G such that number of leaves in T belonging to X is greater than or equal to K if and only if there is a set $S \subseteq X$ such that all of the followings hold true*:

(a) $|X \setminus S| \geqslant K$,
(b) $\langle S \cup Y\rangle$ *is connected, and*
(c) *for any subset* $S' \subseteq S$, $|N_G(S')| \geqslant |S'| + 1$.

Proof. We first prove the "only if part". Suppose there is a spanning tree T such that the number of leaves in T belonging to X is greater than or equal to K. Let $L = \{x \mid x \in X \text{ and } \deg_T(x) = 1\}$. So $|L| \geqslant K$. Now define $S = X \setminus L$. Now it is clear that $|X \setminus S| \geqslant K$. $\langle S \cup Y\rangle$ is connected since all L-nodes are leaves in T and their deletion from T will create a tree spanning $\langle S\cup Y\rangle$. Let us now show that condition (c) holds. Consider any subset $S' \subseteq S$. Let T' be the restriction of tree T to subset $S' \cup N_G(S')$. Then we have

$$\sum_{x\in S'} \deg_T(x) = \bigl|E(T')\bigr| \leqslant \bigl|S' \cup N_G(S')\bigr| - 1$$
$$= |S'| + \bigl|N_G(S')\bigr| - 1.$$

The first equality follows from the fact that each T'-arc has exactly one end point in S'. The inequality is true because T' does not contain any cycles. Thus,

$$\bigl|N_G(S')\bigr| \geqslant \sum_{x\in S'} \bigl(\deg_T(x) - 1\bigr) + 1. \tag{1}$$

Note also that

$$\text{for any } x \in S, \quad \deg_T(x) \geqslant 2$$
$$\text{since } x \text{ is not a leaf in } T. \tag{2}$$

Inequalities (2) and (1) imply that

$$|S'| + 1 \leqslant \sum_{x\in S'} \bigl(\deg_T(x) - 1\bigr) + 1 \leqslant \bigl|N_G(S')\bigr|.$$

Now we need to show the "if part". Assume that the sufficient conditions are satisfied. Let $G' \equiv \langle S \cup Y\rangle$. G' is bipartite with partite sets S and Y. Since G is bipartite, the assumptions about S holds in G' as well. So we must have for any subset $S' \subseteq S$ in G',

$$\bigl|N_G(S')\bigr| \geqslant |S'| + 1 > |S'|.$$

Hence there is a matching M from S to Y by Hall's theorem [3]. Now $F = (S \cup Y, M)$ is a forest where for all $x \in S$, $\deg_F(x) = 1$. Now we show that there exists a forest $\overline{F}$ such that $\deg_{\overline{F}}(x) = 1$ for all $x \in S$ as follows. We show it by an induction on the number of vertices with degree 1 in a forest. Let F' be the forest containing F such that $1 \leqslant \deg_{F'}(x) \leqslant 2$ for all $x \in S$ and let S_1 be the set of all vertices such that $\deg_{F'}(x) = 1$ for all $x \in S_1$. According to assumption we must have $N_G(S_1) > |S_1|$. Hence there must be an edge e joining S_1 and $Y \setminus N_{F'}(S_1)$. So in the forest $F' \cup e$, the number of vertices with degree 1 is fewer than $|S_1|$. Now we can obtain a spanning tree T' of G' when we add some edges to $\overline{F}$ between the connected components of $\overline{F}$. Finally since G is connected and bipartite, for each $x \in X \setminus S$, we can easily add an edge (x, y) of G to T' where $y \in Y$, to get the desired spanning tree T. □

Remark. It is clear that with respect to Problem 2.5, both the conditions stated above (Theorems 2.6 and 2.7) are equivalent to each other, Theorem 2.6 being much simpler. The justification of presenting both the theorems lies partially in the fact that unfortunately we are still unable to settle the issue of complexity for Problem 2.5. However, the two theorems suggest two ways of attacking the problem. As is evident from the proof of Theorem 2.7, it seems that the bipartite maximum matching algorithm might be employed to solve the problem. Also note that the "if part" of the proof of Theorem 2.7 can be replaced by the proof of the "if part" of Theorem 2.6 since the conditions of Theorem 2.7 is just a super set (conditions (a) and (b) of Theorem 2.7 and the conditions stated in Theorem 2.6 are same) of the conditions of Theorem 2.6. Nevertheless, we believe that the rather complicated proof, using the idea of matching, presents a new direction to solve the complexity and algorithmic issue of Problem 2.5.

3. The set version

In this section we first introduce a new notion of "set version" for some well known decision problems. Consider the input instances of Maximum Leaf Spanning Tree problem (Problem 1.3) where we have an integer $K \leqslant |V|$ as a part of the input and we are asked the question whether there exists a spanning tree T in the input graph such that T has K or more leaves. Our notion of set version would pose a similar but different problem where the integer K in the input instance is replaced by a set $X \subseteq V$. To be specific we define the corresponding set version of Problem 1.3 as follows.

Problem 3.1 (*Set Version of Maximum Leaf Spanning Tree problem*). Given a connected graph $G \equiv (V, E)$ and $X \subseteq V$, we are asked the question whether there is a spanning tree T such that $X \subseteq \Pi_T$, where $\Pi_T = \{v \mid v \text{ is a leaf in } T\}$.

Now we present a necessary and sufficient condition for the existence of such a spanning tree as described in Problem 3.1, in the form of Theorem 3.2. As we point out later, the proof of Theorem 3.2 indicates the existence of a polynomial time algorithm to find one such spanning tree as opposed to the NP-completeness of the original problem (Problem 1.3).

Theorem 3.2. *Let $G = (V, E)$ be a connected graph, $X \subseteq V$ and $Y = V \setminus X$. Then there exists a spanning tree T such that $X \subseteq \Pi_T$, if and only if both of the following conditions hold true*:

(1) *$\langle Y \rangle$ is connected.*
(2) *Every X-node has an adjacent node in Y.*

Proof. First we prove the "only if part": Suppose there is a spanning tree T in G such that $X \subseteq \Pi_T$. In that case it is easy to realize that $\langle Y \rangle$ is connected since for all $x \in X$, $\deg_T(x) = 1$ and deleting X does not affect the connectivity of the rest of the graph. The second condition is also trivially true because otherwise two X-nodes would be adjacent in T and thus one of them could not be a leaf of T.

Now we need to prove the "if part": Suppose conditions (1) and (2) are true. First we find a spanning tree T' for $\langle Y \rangle$, then connect each X-node to an adjacent Y-node to get a desired spanning tree T. □

The constructive proof of the sufficient part indicates the existence of a simple polynomial time algorithm to find such a spanning tree. So we in effect get the following theorem.

Theorem 3.3. *Set Version of the Maximum Leaf Spanning Tree problem is polynomially solvable.*

Remark. Note that although the original Maximum Leaf Spanning Tree problem is NP-complete the corresponding set version of the problem can be solved in polynomial time.

Now we consider the set version of Problem 2.5 defined in Section 2 and its complexity issue.

Problem 3.4 (*Set Version of Problem* 2.5). Let G be a connected bipartite graph with partite sets X and Y and $X_1 \subseteq X$. We are asked the question whether there is a spanning tree T_G in G such that $X_1 \subseteq \Pi_T$, where $\Pi_T = \{v \mid v \text{ is a leaf in } T\}$.

Theorem 3.5. *Problem* 3.4 *is polynomially solvable.*

Proof. The proof is simple and makes use of Theorem 2.6. From Theorem 2.6 it is clear that G has a spanning tree as defined in Problem 3.4 if and only if $\langle (X \setminus X_1) \cup Y \rangle$ is connected. Since this can be decided polynomially, the result follows. □

The above discussion may lead us to believe that the corresponding set versions of NP-complete problems are not as hard as the original ones. However, this may not necessarily be the case as follows. We here consider the set version of Problem 2.1, i.e., Minimum Leaf Spanning Tree problem.

Problem 3.6 (*Set Version of Minimum Leaf Spanning Tree problem*). Given a connected graph $G \equiv (V, E)$ and $X \subseteq V$, we are asked the question whether there is a spanning tree T such that $\Pi_T \subseteq X$, where $\Pi_T = \{v \mid v \text{ is a leaf in } T\}$.

By employing the same strategy used to prove Theorem 2.2 (NP-completeness of Minimum Leaf Spanning Tree problem), Problem 3.6 can also be proved to be NP-complete by assuming any $X \subseteq V$ such that $|X| = 2$. So we in effect get the following theorem.

Theorem 3.7. *Set Version of Minimum Leaf Spanning Tree problem is* NP-*complete.*

4. Conclusion and future research

This paper considers the complexity issues of some new and interesting spanning tree problems. In Section 2 we define some new spanning tree problems and present relevant results. In particular we define Minimum Leaf Spanning Tree problem and RLSST problem and prove both of them to be NP-complete. We also consider a variant of Maximum Leaf Spanning Tree problem for bipartite graphs (Problem 2.5) and present two necessary and sufficient conditions. However, we neither prove the problem to be NP-complete nor have presented any polynomial time algorithm to solve it. So the complexity of the problem still remains an open question. In Section 3 we introduce a new notion "set version" and raise the issue of whether the set versions of NP-complete problems are as hard as the original problems. We here prove the set version of Maximum Leaf Spanning Tree problem to be polynomially solvable. It seems that the set versions of spanning tree problems should be generally easier to solve because

(1) In the case of set version, the question about a spanning tree should be answered *only for one set*, and we might have a polynomial time algorithm for answering the question;
(2) While in the original problem we might need to answer the question for *exponential number of sets*, and thus the corresponding algorithm might need exponential time.

However, it is shown here that the idea that set version is easier than the original problem may not necessarily be true in general. It seems that similar studies on set versions of other NP-complete and new problems would be interesting and should be a subject of future research.

References

[1] E.W. Dijkstra, Self-stabilizing systems in spite of distributed control, Comm. ACM 17 (1974) 643–644.
[2] M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman, New York, 1979.
[3] P. Hall, On representation of subsets, J. London Math. Soc. 10 (1935) 26–30.
[4] J.A. Storer, Constructing full spanning trees for cubic graphs, Inform. Process. Lett. 13 (1981) 8–11.