Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Discrete Applied Mathematics ■■■ (■■■■) ■■■–■■■
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Note

A note on the computational complexity of graph vertex partition☆

Yuanqiu Huang[a], Yuming Chu[b]

[a]Department of Mathematics, Hunan Normal University, Changsha 410081, PR China
[b]Department of Mathematics, HuZhou Teacher College, Huzhou, Zhejiang 313000, PR China

Received 23 March 2004; received in revised form 17 March 2006; accepted 5 June 2006

Abstract

A stable set of a graph is a vertex set in which any two vertices are not adjacent. It was proven in [A. Brandstädt, V.B. Le, T. Szymczak, The complexity of some problems related to graph 3-colorability, Discrete Appl. Math. 89 (1998) 59–73] that the following problem is NP-complete: *Given a bipartite graph G, check whether G has a stable set S such that* $G - S$ *is a tree*. In this paper we prove the following problem is polynomially solvable: *Given a graph G with maximum degree* 3 *and containing no vertices of degree* 2, *check whether G has a stable set S such that* $G - S$ *is a tree*. Thus we partly answer a question posed by the authors in the above paper. Moreover, we give some structural characterizations for a graph G with maximum degree 3 that has a stable set S such that $G - S$ is a tree.

Keywords: Graph partition; Stable set; Deficiency number; Polynomial algorithm; Xuong tree

1. Introduction

A *stable set* is a vertex subset of a graph in which any two vertices are not adjacent. Let G be a graph, we use $V(G)$ and $E(G)$ to denote its vertex set and edge set, respectively. For any $X \subseteq V(G) \bigcup E(G)$, $G - X$ denotes the graph obtained from G by deleting all elements in X (note that to delete one vertex in X one must delete all edges incident to it). Vertex-partitions of graphs are closely related to many kinds of graph theoretic problems. For example, checking whether a graph G is k-colorable is equivalent to deciding whether the vertex set of G can be partitioned into k stable sets; a bipartite graph is such a graph whose vertex set can be partitioned into two disjoint stable sets; whereas a split graph is such whose vertex set can be partitioned into a stable set and a clique. Investigating various kinds of vertex-partitions of graphs and also examining the complexity status of the corresponding decision problem have been an interesting topic (for example, see [2–4,8,9]). In particular, Brandstädt et al. in [3] considered the computational complexity of the following decision problem, called STABLE TREE:

Given a graph G, check whether G has a stable set S such that $G - S$ *is a tree*,

and they proved that STABLE TREE is NP-complete, even for a bipartite graph with maximum degree 4. Naturally, the authors in [3] posed the following question:

What is the complexity of STABLE TREE for bipartite graphs with maximum degree $\leqslant 3$?

☆ A Key Project supported by Scientific Research Fund of Hunan Provincial Education Department (05A037), and partly by NSFC (10471039).
E-mail address: hyqq@public.cs.hn.cn (Y. Huang).

0166-218X/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2006.06.003

Please cite this article as: Y. Huang, Y. Chu, A note on the computational complexity of graph vertex partition, Disc. Appl. Math. (2006), doi: 10.1016/j.dam.2006.06.003

Clearly, STABLE TREE is trivial for a graph G with maximum degree $\leqslant 2$, because in this case G is a path or a circuit if G is connected. Thus, in studying the complexity of STABLE TREE for graphs with maximum degree $\leqslant 3$, we are always assuming that the given graph has maximum degree 3.

Motivated by the above problem, in this paper we prove that STABLE TREE is polynomially solvable for a graph G with maximum degree 3 and containing no vertices of degree 2. Thus, we partly answer the question posed by the authors in [3]. Our method is mainly exploiting a result in [6] concerning the time complexity of finding a Xuong tree (its definition will be given in the next section) in a graph.

Meanwhile, we give some structural characterizations for a graph G with maximum degree 3 that has a stable set S such that $G - S$ is a tree.

The graphs in this paper are simple and undirected, and furthermore are connected unless pointed out explicitly. A vertex of a graph G is called a *degree-k* vertex if it has degree k. Denote by $\Delta(G)$ the maximum degree of a graph G. A graph is *unicyclic* if it is connected and contains a unique circuit. For any set X, $|X|$ is the number of elements in X. For notation and terminology not defined here, see [1].

The paper is organized as follows: in the next section we give the definition of a Xuong tree of a graph and also some lemmas. The main results are presented in Section 3.

2. Xoung tree and some elementary lemmas

Let G be a graph, and T be a spanning tree of G. A component F of $G - E(T)$ is called an *odd component* (resp., *even component*) of G with respect to T, if F has odd (resp., even) number of edges. We use the sign $\xi(G, T)$ to denote the number of all odd components of G with respect to T. Then the *deficiency number* of G, denoted by $\xi(G)$, is defined as follows: $\xi(G) = \min_T \xi(G, T)$, where T is taken over all spanning trees of G. Clearly, by the definition $\xi(G) = 0$ if G itself is a tree, and $\xi(G) = 1$ if G is a unicyclic graph. A *Xuong tree* (see [10]) is defined as a spanning tree T of G that satisfies $\xi(G, T) = \xi(G)$. The deficiency number and the Xuong tree of a graph are two important notions in studying the maximum genus of graphs (see [7], for example). Particularly, the time complexity of finding a Xuong tree in a graph is given in [6] (the reader can also see a related paper [5]).

Lemma 1 (*Furst et al. [6]*). *A Xuong tree of a graph G with n vertices and m edges can be constructed in time* $O(m^2 n \log^6 n)$.

Let H be a graph with a degree-k vertex v. Then we say that H is obtained from a graph H' by *adding a degree-k vertex* v, provided that $H' = H - \{v\}$. The following result provides the relationship between the deficiency numbers of these two graphs H and H'.

Lemma 2. *Let H' be a graph and let H be a graph obtained from H' by adding a k-degree vertex v, where $k \geqslant 1$. We have*:

(1) *if k is odd, then* $\xi(H) \leqslant \xi(H')$;
(2) *if k is even, then* $\xi(H) \leqslant \xi(H') + 1$.

Proof. Let T' be a Xuong tree of H', that is, $\xi(H', T') = \xi(H')$. Assume that $e_1, e_2, \ldots, e_k$ are all the edges of H incident with v. Choose $T = T' + \{e_1\}$ as a spanning tree of H, namely, $V(T) = V(T') \bigcup \{v\}$ and $E(T) = E(T') \bigcup \{e_1\}$. We now consider the numbers of odd components of $H - E(T)$ and $H' - E(T')$. We see that the edges $e_2, \ldots, e_k$ must belong to a same component of $H - E(T)$. If k is odd, then the number of odd components of $H - E(T)$ is no more than that of $H' - E(T')$. Thus, by the definition $\xi(H) \leqslant \xi(H, T) \leqslant \xi(H', T') = \xi(H')$. This proves (1). If k is even, (2) is also easily obtained. We only note that in this case the number of odd components of $H - E(T)$ is at most one more than that of $H' - E(T')$. □

For convenience, in the following of this paper, a stable set S of a graph G is said to be a *stable-tree* set, if $G - S$ is a tree.

Lemma 3. *If a graph G has a stable-tree set S, then any subset $S' \subseteq S$ cannot be cut-vertex set of G.*

Please cite this article as: Y. Huang, Y. Chu, A note on the computational complexity of graph vertex partition, Disc. Appl. Math. (2006), doi: 10.1016/j.dam.2006.06.003

Proof. By the definition of the stable-tree set, $G - S$ is a tree and thus is connected. Since G itself is connected, and since S is a stable set of G, it is known that each vertex of S must be adjacent to at least one vertex of $G - S$. Therefore, $G - S'$ is connected. This proves the lemma. □

Lemma 4. *Let G be a graph with $\Delta(G) = 3$. If $\xi(G) = 0$, then G has a stable-tree set S. Moreover, we can choose such S so that S consists of some degree-3 vertices, unless G is a tree itself.*

Proof. If G itself is a tree, the conclusion is trivial. In the following we thus assume that G is not a tree. Let T be a Xuong tree of G, namely $\xi(G, T) = \xi(G) = 0$. Since $\Delta(G) = 3$ and $\xi(G, T) = 0$, each component in $G - E(T)$ is either a path or a circuit with even number of edges (for a path component, it is possibly an isolated vertex). Denote by $\mathscr{F}_{\mathrm{p}}$ the set of all path components of $G - E(T)$, except from isolated vertex components, and by $\mathscr{F}_{\mathrm{c}}$ the set of all circuit components of $G - E(T)$. Because of our assumption that G itself is not a tree, obviously $\mathscr{F}_{\mathrm{p}} \cup \mathscr{F}_{\mathrm{c}} \neq \emptyset$. Now we construct a stable-tree set S of G as follows. First, for each path component $F_{\mathrm{p}} \in \mathscr{F}_{\mathrm{p}}$, since F_{p} has even length, let $F_{\mathrm{p}} = v_1 v_2 \cdots v_{2k} v_{2k+1}$, $k \geqslant 1$, and choose a vertex set

$$\mathscr{S}(F_{\mathrm{p}}) \triangleq \{v_{2i} \mid 1 \leqslant i \leqslant k\}.$$

Since $\Delta(G) = 3$ and T is a spanning tree of G, we easily get the following properties:

(a) $\mathscr{S}(F_{\mathrm{p}})$ is a stable set of G;
(b) Each vertex in $\mathscr{S}(F_{\mathrm{p}})$ has degree two in F_{p}, degree one in T, and degree three in G.

Again, for each circuit component $F_{\mathrm{c}} \in \mathscr{F}_{\mathrm{c}}$, similarly since F_{c} has even length, let $F_{\mathrm{c}} = u_1 u_2 \cdots u_{2\ell} u_1$, $\ell \geqslant 2$, and choose a vertex set

$$\mathscr{S}(F_{\mathrm{c}}) \triangleq \{u_{2i} \mid 1 \leqslant i \leqslant \ell\}.$$

Similarly, we have the following properties:

(c) $\mathscr{S}(F_{\mathrm{c}})$ is a stable set of G;
(d) Each vertex in $\mathscr{S}(F_{\mathrm{c}})$ has degree two in F_{c}, degree one in T, and degree three in G.

Now we take

$$S \triangleq \left(\bigcup_{F_{\mathrm{p}} \in \mathscr{F}_{\mathrm{p}}} \mathscr{S}(F_{\mathrm{p}}) \right) \cup \left(\bigcup_{F_{\mathrm{c}} \in \mathscr{F}_{\mathrm{c}}} \mathscr{S}(F_{\mathrm{c}}) \right).$$

We shall prove that S is as desired in the lemma. First, by properties (b) and (d) above, each vertex in S has degree one in T, and thus $G - S$ is connected. Furthermore, by the choice of S we know that $E(G - S) \subseteq E(T)$. So, $G - S$ is a tree. In order to prove that S is a stable set of G, we only prove that, for any two vertices $x, y \in S$, x and y are not adjacent in G. By contradiction, assume that e is an edge of G that joins x and y. Let $F_1, F_2 \in \mathscr{F}_{\mathrm{p}} \cup \mathscr{F}_{\mathrm{c}}$ be the components of $G - E(T)$ that contain x and y, respectively. Combining the choice of S and the properties (a) and (c) above, we can get that $F_1 \neq F_2$. Since $F_1 \neq F_2$, it follows that $e \in E(T)$. Since $\Delta(G) = 3$, by the properties (b) and (d) above we can conclude that e does not connect any other edges of T, contradicting that T is a spanning tree of G. Thus, S is a stable set of G. Finally, properties (b) and (d) above ensure that each vertex of S has degree three in G. Thereby, the proof of the lemma is obtained. □

3. The main results

In this section we display our main results. The following first theorem provides a necessary and sufficient condition for a graph with $\Delta(G) = 3$ and containing no degree-2 vertices that has a stable-tree set.

Theorem 5. *Let G be a graph with $\Delta(G) = 3$ and containing no degree-2 vertices. Then G has a stable-tree set S if and only if $\xi(G) = 0$.*

Please cite this article as: Y. Huang, Y. Chu, A note on the computational complexity of graph vertex partition, Disc. Appl. Math. (2006), doi: 10.1016/j.dam.2006.06.003

Proof. Clearly, the necessity follows directly from Lemma 4. We only prove the sufficiency. Let S be a stable-tree set of G. Since G has no degree-2 vertex, each vertex in S has degree one or three in G. Again, since S is a stable-tree set of G, we know that $G - S$ is a tree, and thus $\xi(G - S) = 0$ by the definition. On the other hand, we note that G can be obtained from $G - S$ by repeatedly adding all degree-1 degree-3 vertices in S. Therefore, repeatedly applying Lemma 2(1) we get that $\xi(G) \leqslant \xi(G - S) = 0$. Because $\xi(G)$ is a nonnegative integer, it implies that $\xi(G) = 0$. This proves the sufficiency. □

Using Theorem 5, we now give a polynomial algorithm for STABLE TREE for a given graph G with $\Delta(G) = 3$ and containing no degree-2 vertices. The algorithm is simple.

Algorithm (deciding whether a given graph G with $\Delta(G) = 3$ and containing no degree-2 vertices has a stable-tree set):

1. Construct a Xuong tree T of G.
2. For each component F of $G - E(T)$, count the number of edges of F.
3. Compute $\xi(G)$, that is, determine the number of components of $G - E(T)$ with odd number edges.
4. If $\xi(G) = 0$, the answer is YES, otherwise NO.

The correctness of the algorithms follows directly from Theorem 5, and its time complexity is mainly determined by Step 1, which runs in $O(m^2 n \log^6 n)$ time by Lemma 1, where m and n are, respectively, the number of edges and vertices of G. Therefore we have the following theorem.

Theorem 6. *STABLE TREE is polynomially solvable for a given graph G with $\Delta(G) = 3$ and containing no degree-2 vertices.*

Thus, by Theorem 6 we partly answer the question posed by the authors in [3]. Furthermore, in our result we remove the "bipartite" restriction for the given graph.

We see that Theorem 5 gives a necessary and sufficient condition for a graph with $\Delta(G) = 3$ and containing no degree-2 vertices that has a stable-tree set. If we delete the restriction "containing no degree-2 vertices", then we have the following result.

Theorem 7. *Let G be a graph with $\Delta(G) = 3$, and let $\mathcal{N}_2(G)$ denote the set of all the degree-2 vertices of G. Then G has a stable-tree set S, if and only if there exists a subset $X \subseteq \mathcal{N}_2(G)$ satisfying the following conditions*:

(1) *X is a stable set of G*;
(2) *$G - X$ is connected*;
(3) $\xi(G - X) = 0$.

Proof. First let us prove the necessity. Assume that G has a stable-tree set S. Since $\Delta(G) = 3$, we can write S as the disjoin union: $S = S_1 \bigcup S_2 \bigcup S_3$, where S_i consists of some i-degree vertices of G ($i = 1, 2, 3$). Because S is a stable-tree set of G, $G - S$ is a tree, and so $\xi(G - S) = 0$ by the definition. Take $X = S_2$. Obviously, $X \subseteq \mathcal{N}_2(G)$. We shall prove that X satisfies conditions (1)–(4) of the theorem. First, X is stable set of G, because so is S. This is condition (1). Condition (2) follows from Lemma 3 and the fact that S is a stable-tree set of G and $X \subseteq S$. Note that $S_1 \cup S_3$ is also a stable set of G. We see that the graph $G - X$ can be obtained from $G - S$ by successively adding all the vertices in $S_1 \cup S_3$. Since each vertex in $S_1 \cup S_3$ has degree one or three in G, repeatedly using Lemma 2(1) we get that $\xi(G - X) \leqslant \xi(G - S) = 0$, implying that $\xi(G - X) = 0$. This is condition (3). This proves the necessity.

Now we prove the sufficiency. Assume that there exists $X \subseteq \mathcal{N}_2(G)$ such that all conditions (1)–(4) of the theorem are satisfied. By condition (2), $G - X$ is connected, and we consider two cases.

Case 1: $G - X$ is a tree. Then take $S = X$. By condition (1) we know that S is a stable-tree set of G.

Case 2: $G - X$ is not a tree. Since $\Delta(G) = 3$ $G - X$ is connected, $2 \leqslant \Delta(G - X) \leqslant 3$, and thus there are two subcases.

Subcase 2.1: if $\Delta(G - X) = 2$. Combining condition (2) with the assumption that $G - X$ is not a tree, we see that $G - X$ is a unicyclic graph, and thus $\xi(G - X) = 1$ by the definition. This contradicts to condition (3). So, this subcase is impossible to happen.

Subcase 2.2: $\Delta(G-X)=3$. In this subcase, it follows from conditions (2),(3), and Lemma 4 that $G-X$ has a stable-tree set S'. Furthermore, S' consists of some degree-3 vertices of $G-X$. Note that S' is also a stable set of G. Taking $S=X\cup S'$, we shall prove that S is a stable-tree set of G. Clearly, $X\cap S'=\emptyset$. Since S' is a stable-tree set of $G-X$, we have that $(G-X)-S'=G-(X\cup S')$ is a tree. That is to say, $G-S$ is a tree. Note that $G-S=(G-X)-S'$ is a tree, and that both X and S' are stable sets of G. Moreover, as every vertex v in S' has degree $\triangle(G)=3$, v is nonadjacent to any vertex in X, hence $S=X\cup S'$ is a stable set of G. By the arguments in this subcase, S is a stable-tree set of G.

By the above covered cases the proof of the sufficiency is complete. □

Note: Applying Theorem 5, we give a polynomial-time algorithm for STABLE TREE for a given G with $\Delta(G)=3$ and containing no degree-2 vertices. However, presently we are not able to find a polynomial-time algorithm based on Theorem 7 for STABLE TREE for a given G with $\Delta(G)=3$ and containing some degree-2 vertices. Thus the complexity status of the problem STABLE TREE is still open for this case.

Acknowledgments

We would like to thank the anonymous referees for their helpful suggestions and pointing out some mistakes in the first version of this paper.

References

[1] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, North-Holland, New York, 1976.
[2] A. Brandstädt, Partition of graphs into one or two independent sets and cliques, Discrete Math. 152 (1996) 47–54 (Corrigendum: Discrete Math. 186 (1998) 195).
[3] A. Brandstädt, V.B. Le, T. Szymczak, The complexity of some problems related to graph 3-colorability, Discrete Appl. Math. 89 (1998) 59–73.
[4] K. Bryś, Z. Lonc, Clique and anticlique partitions of graphs, Discrete Math. 185 (1998) 41–49.
[5] J. Chen, S.P. Kanchi, Graph ear decompositions and graphs embeddings, SIAM J. Discrete Math. 12 (2) (1999) 229–242.
[6] M. Furst, J.L. Gross, A. McGeoch, Finding a maximum genus graph imbedding, J. Assoc. Comput. Mach. 35 (3) (1988) 523–534.
[7] J.L. Gross, T.W. Tucker, Topological Graph Theory, Wiley-Interscience, New York, 1984.
[8] R. Mosca, Polynomial algorithms for the maximum stable set problem on particular classes of P_5-free graphs, Inform. Process. Lett. 61 (1997) 137–143.
[9] K. Wada, A. Takaki, K. Kawaguchi, Efficient algorithms for a mixed k-partition problem of graphs without specifying bases, Theoret. Comput. Sci. 201 (1998) 233–248.
[10] N.H. Xuong, How to determine the maximum genus of a graph, J. Combin. Theory Ser. B 26 (1979) 217–225.