ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
Discrete Applied Mathematics 129 (2003) 371–382
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

A polynomial algorithm for finding T-span of generalized cacti

Krzysztof Giaro, Robert Janczewski*,[1], Michał Małafiejski

Foundations of Informatics Department, Faculty of Electronics, Telecommunications and Informatics, Technical University of Gdańsk ul. Narutowicza 11/12, Gdańsk, Poland

Received 31 October 2000; received in revised form 18 June 2002; accepted 29 July 2002

Abstract

It has been known for years that the problem of computing the T-span is NP-hard in general. Recently, Giaro et al. (Discrete Appl. Math., to appear) showed that the problem remains NP-hard even for graphs of degree $\Delta \leqslant 3$ and it is polynomially solvable for graphs with degree $\Delta \leqslant 2$. Herein, we extend the latter result. We introduce a new class of graphs which is large enough to contain paths, cycles, trees, cacti, polygon trees and connected outerplanar graphs. Next, we study the properties of graphs from this class and prove that the problem of computing the T-span for these graphs is polynomially solvable.

Keywords: T-coloring; T-span; Cactus; Outerplanar graph

1. Introduction

In order to establish some notation and terminology that will be necessary later on, let us recall that if T is a T-*set*, i.e. a finite set of non-negative integers such that $0 \in T$, then a T-*coloring* of a given simple graph $G=(V,E)$ is any function $c: V \to \mathbb{Z}$ such that $|c(u)-c(v)| \notin T$ whenever $\{u,v\} \in E$. The T-*span* of G, denoted by $\mathrm{sp}_T(G)$, is the minimum span over all T-colorings of G, where the *span* of a T-coloring c of G is the distance between the largest and the smallest color used by c.

* Corresponding author.
E-mail addresses: giaro@eti.pg.gda.pl (K. Giaro), skalar@eti.pg.gda.pl (R. Janczewski), mima@eti.pg.gda.pl (M. Małafiejski).
[1] Supported by FNP.

0166-218X/03/$ - see front matter
PII: S0166-218X(02)00575-9

T-colorings and T-span are strictly connected with ordinary vertex colorings and the chromatic number of graphs. For instance, it is known [1] that if set T is r-initial, i.e. $\{0,1,\dots,r\}\subseteq T$ and T contains no positive multiples of $r+1$, then $\mathrm{sp}_T(G)=(r+1)(\chi(G)-1)$, where $\chi(G)$ is the chromatic number of G. Furthermore, Raychaudhuri [7] showed that if T is a k multiple of s set, i.e. $T\subseteq\{0,s,s+1,s+2,\dots,ks\}$ and $\{0,s,2s,\dots,ks\}\subseteq T$, then

$$\mathrm{sp}_T(G)=\begin{cases}(k+1)\chi(G)-ks-1 & \text{if } s|\chi(G),\\ ks\lfloor\frac{\chi(G)}{s}\rfloor+\chi(G)-1 & \text{otherwise.}\end{cases}$$

T-colorings and T-span have been studied for two decades, mostly because of their potential application in the channel assignment problem (refer to [4,8] for details). A number of their properties have been discovered. We need some of them so let us recall that the problem of computing the T-span is NP-hard in the strong sense even for complete graphs (see [5] for the proof of NP-hardness and [3] for the proof of NP-hardness in the strong sense) as well as r-regular ($r\geqslant 3$) and subcubic graphs (see [2]). Of course, there are several cases where considered problem becomes polynomial. The problem is easy for bipartite graphs since it is known that if G is a bipartite graph, then

$$\mathrm{sp}_T(G)=\begin{cases}m_T & \text{if } G \text{ is not edgeless,}\\ 0 & \text{otherwise,}\end{cases}\tag{1}$$

where m_T is the smallest positive integer not belonging to T. Giaro et al. [2] showed that the problem of computing the T-span of graphs with degree not exceeding two is polynomial. Herein, we generalize this result.

The remainder of the paper is organized as follows. Section 2 contains some preliminary definitions and results. We introduce a notion of a CP-decomposition of a graph and define thorny graphs. Next, we discuss some of their properties. A number of examples are given in the third section. We prove that every path, cycle, cactus, tree, polygon tree and connected outerplanar graph is a thorny graph. A polynomial time algorithm for finding the T-span of thorny graphs is given in the final section of this paper.

2. Preliminaries

Let us recall that if G is a simple graph then symbols $V(G), E(G), m(G)$ and $n(G)$ denote the set of vertices, the set of edges, the number of edges and the number of vertices of G, respectively. We shall drop the reference to the graph, if G is clear from context.

Definition 1. A *union* of graphs G and H, denoted by $G\cup H$, is the graph satisfying $V(G\cup H)=V(G)\cup V(H)$ and $E(G\cup H)=E(G)\cup E(H)$.

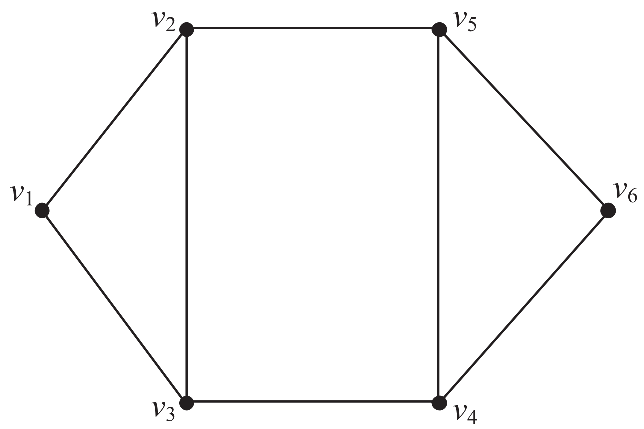

Fig. 1. An example of a thorny graph.

Definition 2. An *intersection* of graphs G, H which are not vertex disjoint, denoted by $G \cap H$, is the graph satisfying $V(G \cap H) = V(G) \cap V(H)$ and $E(G \cap H) = E(G) \cap E(H)$.

Let us note that the union and intersection of simple graphs have almost the same properties as union and intersection of sets—these operations are all associative and commutative, for example. Moreover, the union of connected graphs which are not vertex disjoint is connected and the following equality:

$$\chi(G \cup H) = \max\{\chi(G), \chi(H)\} \tag{2}$$

holds if simple graphs G, H are vertex disjoint or their intersection is a path with at most two vertices. These simple observations will be used later to prove some of the properties of thorny graphs.

Definition 3. A sequence $G_1, G_2, \ldots, G_k$ of subgraphs of a given graph G is said to be a *CP-decomposition* of G of length k iff the following conditions are fulfilled

(1) all graphs in the sequence are cycles or paths,
(2) $(G_1 \cup \cdots \cup G_{i-1}) \cap G_i$ is K_1 or K_2 for each $i = 2, 3, \ldots, k$,
(3) $G = G_1 \cup \cdots \cup G_k$.

Definition 4. A graph G is said to be *thorny* iff it has at least one CP-decomposition.

Every cycle and every path are thorny graphs because every path and cycle has a CP-decomposition of length one. Also, the graph G of Fig. 1 has a CP-decomposition of length three, e.g. G_1, G_2, G_3 where $G_1 = G(\{v_1, v_2, v_3\})$, $G_2 = G(\{v_2, v_3, v_4, v_5\})$ and $G_3 = G(\{v_4, v_5, v_6\})$. Lots of other examples will be given in the next section. Now we restrict our attention to the properties of CP-decompositions and thorny graphs.

Proposition 5. *Every thorny graph is connected, planar and tripartite.*

Proof. Connectivity and planarity of thorny graphs follow directly from their definition. Tripartiteness will be proved by induction with a little help of equality (2). Let G be a thorny graph and let $G_1, G_2, \dots, G_k$ be any of its CP-decompositions. Let $H_1, H_2, \dots, H_k$ be a sequence of graphs defined as follows:

$$H_i = G_1 \cup G_2 \cup \cdots \cup G_i, \quad i = 1, 2, \dots, k.$$

We will use induction on i to show that every graph H_i, $i = 1, 2, \dots, k$ is tripartite; it will complete the proof since $G = H_k$. The claim is obvious for $i = 1$, so to complete the proof it suffices to note that if graph H_i $(i < k)$ is tripartite then graph H_{i+1} is tripartite, which follows from equality (2) and the fact that graph H_{i+1} is a union of tripartite graphs H_i, G_{i+1} whose intersection is K_1 or K_2. □

In the following we will denote by $k(G)$ the minimum length over all possible CP-decompositions of a given thorny graph G. The reader can easily verify that cycles and paths are the only thorny graphs satisfying $k(G) = 1$ and if G is the graph shown in Fig. 1, then $k(G) = 3$.

Proposition 6. *If G is a thorny graph with at least two vertices, then*

$$m(G) \leqslant 2n(G) - 3. \tag{3}$$

Proof. Let $G_1, G_2, \dots, G_k$ be a CP-decomposition of G such that $k = k(G)$ and let $H_1, H_2, \dots, H_k$ be a sequence of graphs defined as follows:

$$H_i = G_1 \cup G_2 \cup \cdots \cup G_i, \quad i = 1, 2, \dots, k.$$

We will use induction on i to show that every graph H_i, $i = 1, 2, \dots, k$ satisfies inequality $m(H_i) \leqslant 2n(H_i) - 3$; it will complete the proof since $G = H_k$.

We begin by observing that all graphs in the CP-decomposition under consideration must have at least two vertices and therefore $m(G_i) \leqslant 2n(G_i) - 3$ for each $i = 1, 2, \dots, k$. Assume that $i < k$ and $m(H_i) \leqslant 2n(H_i) - 3$. There are only two cases to consider.

(1) $G_{i+1} \cap H_i$ is K_1. Then $n(H_{i+1}) = n(H_i) + n(G_{i+1}) - 1$, $m(H_{i+1}) = m(H_i) + m(G_{i+1})$ and thus

$$m(H_{i+1}) \leqslant 2n(H_i) - 3 + 2n(G_{i+1}) - 3 = 2n(H_{i+1}) - 4.$$

(2) $G_{i+1} \cap H_i$ is K_2. Then $n(H_{i+1}) = n(H_i) + n(G_{i+1}) - 2$, $m(H_{i+1}) = m(H_i) + m(G_{i+1}) - 1$ and thus

$$m(H_{i+1}) \leqslant 2n(H_i) - 3 + 2n(G_{i+1}) - 3 - 1 = 2n(H_{i+1}) - 3. \quad \square$$

Let G_1 be the path satisfying the equality $V(G_1) = \{1, 2\}$ and G_n $(n \geqslant 2)$ be the cycle satisfying the equality $V(G_n) = \{n-1, n, n+1\}$. All graphs in the sequence T_n defined recursively as follows:

$$T_n = \begin{cases} G_1 & \text{if } n = 2, \\ T_{n-1} \cup G_{n-1} & \text{if } n > 2. \end{cases}$$

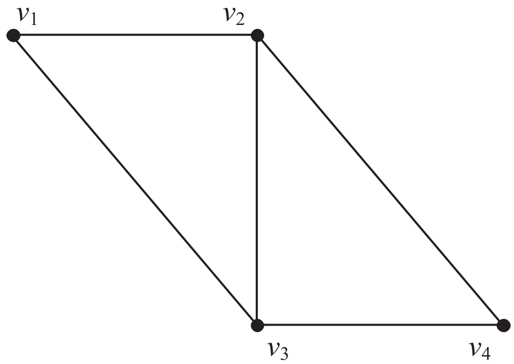

Fig. 2. The graph T_4.

are thorny (T_3 is a cycle with three vertices and T_4 is shown in Fig. 2). The reader can easily verify that $n(T_n) = n$ and $m(T_n) = 2n - 3$ so the upper bound (3) is tight.

Proposition 7. *Let G be a thorny graph. If H is an induced subgraph of G then H is thorny if and only if H is connected.*

Proof. It suffices to show that if H is an induced and connected subgraph of G then H is a thorny graph. The implication is obvious whenever $k(G) = 1$. Assume that the implication is true for all thorny graphs with $k(G) < p$ ($p \geqslant 2$). Let G be a thorny graph with $k(G) = p$ and $G_1, G_2, \ldots, G_p$ be a CP-decomposition of G. If H is a subgraph of G_p or a subgraph of $G' = G_1 \cup G_2 \cup \cdots \cup G_{p-1}$, then the claim is obvious. If H is neither a subgraph of G_p nor a subgraph of G', then $V(H) \cap V(G_p) \neq \emptyset$ and $V(H) \cap V(G') \neq \emptyset$. The reader can verify that graphs $H_1 = H \cap G'$ and $H_2 = H \cap G_p$ are connected. H_2 must be a path or cycle as a connected subgraph of a cycle or path G_k and H_1 must be thorny as a connected and induced subgraph of a thorny graph G' with $k(G') < p$. Let $H_1', H_2', \ldots, H_k'$ be a CP-decomposition of H_1. To complete the proof, it is sufficient to note that $H_1', H_2', \ldots, H_k', H_2$ is a CP-decomposition of H.

To express other properties of thorny graphs we must mention the definition of the core of graph. Note that the word ‘induced’ cannot be removed from the text of Proposition 7 (see Fig. 3 for a counter-example).

Definition 8. The *core* of a graph G is a subgraph obtained by pruning away all pendant vertices successively until there is no vertex of degree one.

Proposition 9. *Let G be a simple graph and H be its core. Then G is thorny if and only if H is thorny.*

Proof. ($\Rightarrow$) The core of any connected graph is a connected and induced subgraph of the graph. The claim follows now from Proposition 7.

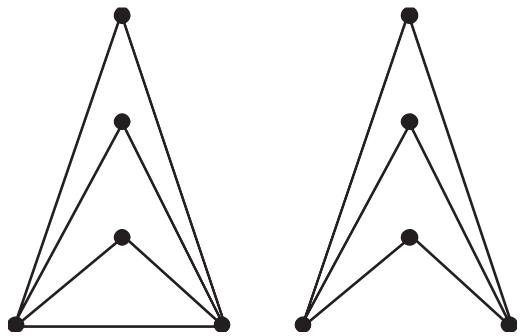

Fig. 3. An example of a thorny graph (left) and its subgraph (right) that is not thorny.

($\Leftarrow$) A graph obtained from a thorny graph by adding one pendant vertex is thorny. Thus, G is thorny as a result of adding pendant vertices one after the other to thorny graph H. □

In the following, we will say that connected graph G that is not a cycle has a *pendant* cycle iff there is such a graph H and a cycle C that $G = H \cup C$ and $H \cap C$ is K_1 or K_2.

Proposition 10. *Every thorny graph other than a cycle has a pendant vertex or a pendant cycle.*

Proof. It follows immediately from the definition of a pendant cycle and the definition of thorny graphs. □

Corollary 11. *Every thorny graph has a vertex of degree less than or equal to* 2.

In general, it may be hard to decide whether a given graph belongs to a given class or not. It appears that deciding whether a given graph is thorny or not is easy. Moreover, we can easily find a CP-decomposition of any thorny graph. The reader can verify that the following algorithm tests whether a given graph G is thorny and if so then it returns one of its CP-decompositions.

(1) Let list $L = \emptyset$. If G is not connected, then stop.
(2) If G is edgeless or it is a cycle, then make G the head of L and stop.
(3) If G has nor a pendant vertex neither a pendant cycle, then let $L = \emptyset$ and stop.
(4) If G has a pendant vertex v then there are graphs G_1 and G_2 such that $G = G_1 \cup G_2$, $V(G_1 \cap G_2) = \{u\}$ and G_2 is a path with two vertices u and v, where u is the only neighbor of v. Make G_2 the head of L, let $G = G_1$ and go to Step 2.
(5) If G has a pendant cycle, i.e. there is a cycle C and a graph H such that $G = H \cup C$ and all other requirements are satisfied then make C the head of L, let $G = H$ and go to Step 2.

The algorithm returns a list L which is non-empty if graph G is thorny. If the list is not empty then it contains one of CP-decompositions of G.

The above algorithm performs at most n steps. In each step it finds a pendant vertex or pendant cycle of G, removes some vertices from G and adds a new item to list L. It is obvious that we are able to find all pendant vertices of G, remove some vertices from G and add a new item to L in $O(n^2)$ time. It appears that finding pendant cycles of G is also realizable in $O(n^2)$ time since to find them it suffices to build a subgraph of G induced by vertices of degree 2 and for each connected component of the subgraph (these components must be paths) verify whether their endpoints have common neighbor or adjacent neighbors in the rest of G. Thus, the computational complexity of the algorithm is $O(n^3)$.

3. Examples

Now we are ready to prove that many well-known graphs are thorny. Let us recall that a simple connected graph G is said to be a cactus iff every two different cycles in G have at most one vertex in common.

Proposition 12. *Every cactus is a thorny graph.*

Proof. Since the only edgeless cactus is thorny, we may assume that a given cactus G has at least one edge. Let $S = \{G_1, G_2, \dots, G_k\}$ be a set containing all subgraphs of G which are cycles or paths with only one edge being a bridge in G. Let $\pi : \{1, 2, \dots, k\} \to \{1, 2, \dots, k\}$ be a function defined recursively as follows:

$$\pi(i) = \begin{cases} 1 & \text{if } i = 1, \\ \min\{r \notin \pi(\{1, 2, \dots, i-1\}) : V(G_r) \cap V(H_{i-1}) \neq \emptyset\} & \text{if } 2 \leqslant i \leqslant k, \end{cases}$$

where $H_i = G_{\pi(1)} \cup G_{\pi(2)} \cup \cdots \cup G_{\pi(i)}$. Since G is connected, π is a well-defined one-to-one function. We will use induction on i to show that $G_{\pi(1)}, G_{\pi(2)}, \dots, G_{\pi(i)}$ is a CP-decomposition of H_i for each $i = 1, 2, \dots, k$; it will complete the proof since $G = H_k$.

Clearly, the claim is true for $i = 1$. Assume that the claim holds for i $(i < k)$. To prove that it holds for $i + 1$ it suffices to show that graphs $H_i, G_{\pi(i+1)}$ have exactly one vertex in common. There are only two cases to consider.

(1) $G_{\pi(i+1)}$ is a path, i.e. the only edge $\{u, v\}$ of $G_{\pi(i+1)}$ is a bridge in G. If both vertices u, v were also vertices of H_i then there would be two different paths connecting u, v in G—bridge $\{u, v\}$ and a path connecting u, v in H_i (such a path exists since H_i is connected)—a contradiction. Thus, the graphs have exactly one vertex in common.

(2) $G_{\pi(i+1)}$ is a cycle. G is a cactus so graphs $G_1, G_2, \dots, G_k$ must be edge disjoint and therefore graphs $G_{\pi(i+1)}, H_i$ are edge disjoint. If graphs H_i and $G_{\pi(i+1)}$ had at least

two common vertices u, v then there would be at least three edge disjoint paths connecting these vertices (two paths in $G_{\pi(i+1)}$ and at least one in H_i; these paths are edge disjoint, since graphs $G_{\pi(i+1)}, H_i$ are edge disjoint)—a contradiction. Thus the graphs have exactly one vertex in common. □

Proposition 13. *Every connected outerplanar graph is thorny.*

Proof. Throughout the proof, $S(G)$ denotes a set containing all subgraphs of a connected outerplanar graph G which are cycles surrounding faces or paths with exactly one edge being a bridge in G. We will use induction on the cardinality of $S(G)$ to show that elements of $S(G)$ can be put in a sequence being a CP-decomposition of G.

The claim is true for paths with at most two vertices and cycles, i.e. connected outerplanar graphs satisfying $|S(G)| \leqslant 1$. Assume that it holds for graphs satisfying $|S(G)| = k$ $(k \geqslant 1)$ and let us check whether it holds for graphs with $|S(G)| = k + 1$. Let G be such a connected outerplanar graph that $S(G) = \{G_1, G_2, \ldots, G_{k+1}\}$. There are only two cases to consider.

(1) Every two different elements of $S(G)$ have at most one common vertex. Then G must be a cactus and the claim follows from the proof of Proposition 12.

(2) There are such numbers i, j that $1 \leqslant i < j \leqslant k + 1$ and the intersection of graphs G_i, G_j is a path with two vertices. Without loss of generality we assume that $i = k$ and $j = k + 1$. Let e be the only edge of $G_k \cap G_{k+1}$. Since G_k and G_{k+1} must be cycles, the graph H obtained from G by removing edge e must be a connected outerplanar graph. Furthermore, $S(H) = \{H_1, H_2, \ldots, H_k\}$, where $H_i = G_i$ for $i = 1, 2, \ldots, k - 1$ and H_k is a cycle obtained from $G_k \cup G_{k+1}$ by removing e. By the inductional hypothesis, there is such a one-to-one function $\pi : \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}$ that $H_{\pi(1)}, H_{\pi(2)}, \ldots, H_{\pi(k)}$ is a CP-decomposition of H. Let $l = \pi^{-1}(k)$. The reader can easily verify that if we replace H_k in the decomposition by G_k, G_{k+1} (if $H_k \cap (H_{\pi(1)} \cup H_{\pi(2)} \cup \cdots \cup H_{\pi(l-1)})$ is a subgraph of G_k) or G_{k+1}, G_k (otherwise), then we obtain a CP-decomposition of G. □

Proposition 14. *Every tree of polygons is a thorny graph.*

Proof. It is clear since they may be defined as the smallest class of graphs $\mathscr{G}$ such that (1) cycles are in $\mathscr{G}$ and (2) if $G \in \mathscr{G}$ then $G \cup H \in \mathscr{G}$ for all cycles H such that $G \cap H$ is K_2.

4. The algorithm

To show how to compute efficiently the T-span of a thorny graph, we have to introduce the notion of odd girth of a graph, prove that it can be found in polynomial time, write some words about the so-called T-graphs and solve the problem of computing the T-span of odd cycles. We begin with the definition of odd girth.

Definition 15. Let G be any non-bipartite graph. The *odd girth* of G, denoted by $\mathrm{og}(G)$, is the minimum number of vertices over all odd cycles being subgraphs of G.

It is well known that the distance between any vertices u, v of G, denoted here by $d_G(u, v)$, may be computed in $\mathrm{O}(m+n)$ time. It appears that the odd girth of any non-bipartite graph may be found in polynomial time, too. The theorem given below tells us that it can be done in $\mathrm{O}(n^2+mn)$ time.

Theorem 16. *Let G be a non-bipartite graph and H be defined by $V(H) = V(G) \times \{1, 2\}$ and $E(H) = \{\{(u, 1), (v, 2)\}, \{(v, 1), (u, 2)\} : \{u, v\} \in E(G)\}$. Then*

$$\mathrm{og}(G) = \min\{d_H((v, 1), (v, 2)) \colon v \in V(G)\}$$

Proof. Let $k = \mathrm{og}(G)$ and $l = \min\{d_H((v, 1), (v, 2)) \colon v \in V(G)\}$. Throughout the proof, the phrase "a sequence $v_0, v_1, \ldots, v_{k-1}$ of vertices constitutes a path (a cycle)" means that all vertices in the sequence are different and v_i is a neighbor of v_{i+1} for each $i = 0, 1, \ldots, k-2$ (v_i is a neighbor of $v_{(i+1) \bmod k}$ for each $i = 0, 1, \ldots, k-1$).

By definition of k, there is a sequence $v_0, v_1, \ldots, v_{k-1}$ of vertices of G which constitutes an odd cycle. It is easy to check that the sequence $(v_0, 1), (v_1, 2), \ldots, (v_{k-1}, 1), (v_0, 2)$ constitutes a path in H. Hence $l \leqslant d_H((v_0, 1), (v_0, 2)) \leqslant k$.

Let $w_0 = (u_0, 1), w_1 = (u_1, 2), \ldots, w_{l-1} = (u_{l-1}, 1), w_l = (u_0, 2)$ be a sequence of vertices of H that constitutes a path of length l. Clearly, the number l must be odd and vertices $u_i, u_{(i+1) \bmod l}$ must be adjacent in G for each $i = 0, 1, \ldots, l-1$. We will show that all vertices in the sequence $u_0, u_1, \ldots, u_{l-1}$ are different; it will complete the proof since if it holds then $u_0, u_1, \ldots, u_{l-1}$ constitutes an odd cycle in G, which yields $k \leqslant l$.

Suppose that there are indices $i, j \in \{0, 1, \ldots, l-1\}$ for which $i < j$ and $u_i = u_j$. Then the set of positive integers k such that $u_i = u_{i+k}$ for some $i \in \{0, 1, \ldots, l-1\}$ is non-empty. Let k_0 denote its minimum and i_0 be such an integer that $u_{i_0} = u_{i_0+k_0}$. Since adjacent vertices cannot be equal, we see that $k_0 > 1$. Moreover, k_0 cannot be even since otherwise $w_{i_0} = w_{i_0+k_0}$. All vertices in the sequence $u_{i_0}, u_{i_0+1}, \ldots, u_{i_0+k_0-1}$ must be different so the sequence $w_{i_0}, w_{i_0+1}, \ldots, w_{i_0+k_0}$ constitutes a path connecting vertices $(u_{i_0}, 1), (u_{i_0}, 2)$. The length of the path is $k_0 < l$, which is a contradiction to the definition of l. $\square$

It is known that there is a relation among the T-span, homomorphisms of graphs and T-graphs. Liu [6] showed that if d is a positive integer then $\mathrm{sp}_T(G) \leqslant d-1$ iff $G \to G_T^d$, where G_T^d is the graph satisfying

$$V(G_T^d) = \{0, 1, \ldots, d-1\}, \quad E(G_T^d) = \{\{u, v\} : |u - v| \notin T\}$$

and the symbol $G \to H$ means that there is a homomorphism from G to H, i.e. a function $h : V(G) \to V(H)$ such that $\{h(u), h(v)\} \in E(H)$ whenever $\{u, v\} \in E(G)$. G_T^d is called the T-graph of order d and it was introduced by Liu. It appears that the odd girth of T-graphs can be found in $\mathrm{O}(d^2)$ time.

Theorem 17. *Let d be such a positive integer that graph G_T^d is not bipartite and H be defined by $E(H)=\{\{(u,1),(v,2)\},\{(v,1),(u,2)\} : \{u,v\}\in E(G_T^d)\}$ and $V(H)=V(G_T^d)\times\{1,2\}$. Then*

$$\mathrm{og}(G_T^d)=d_H((0,1),(0,2)).$$

Proof. Let us note that construction transforming G_T^d into H has the following nice property—it changes cycles of odd length into paths of the same odd length (see the proof of Theorem 16). The claim now follows from Theorem 16 and the fact that 0 is a vertex of one of the shortest odd-length cycles in G_T^d. □

Theorem 18. *If d is a positive integer, then $\mathrm{sp}_T(C_{2n+1})\leqslant d-1$ if and only if G_T^d is not bipartite and $\mathrm{og}(G_T^d)\leqslant 2n+1$.*

Proof. ($\Leftarrow$) First, observe that if $k\leqslant n$ then $C_{2n+1}\to C_{2k+1}$. Hence $C_{2n+1}\to G_T^d$ since G_T^d contains an odd cycle with the number of vertices that is less than or equal to $2n+1$. Thus $\mathrm{sp}_T(C_{2n+1})\leqslant d-1$.

($\Rightarrow$) The graph G_T^d cannot be bipartite since $C_{2n+1}\to G_T^d$. Moreover, homomorphic image of a cycle C_{2n+1} must contain an odd cycle with the number of vertices less than or equal to $2n+1$. Therefore $\mathrm{og}(G_T^d)\leq 2n+1$. □

As we mentioned in the Introduction, the problem of computing the T-span of odd cycles had been solved—Giaro et al. invented algorithm for finding $\mathrm{sp}_T(C_{2n+1})$ that is of complexity $\mathrm{O}(n|T|^2\log|T|)$. Below we improve their result by introducing an algorithm for finding $\mathrm{sp}_T(C_{2n+1})$ with complexity $\mathrm{O}(|T|^2\log|T|)$.

Let us observe that to verify whether $\mathrm{sp}_T(C_{2n+1})\leqslant d-1$, it suffices to do the three following steps: (1) build G_T^d, (2) check whether G_T^d is not bipartite and (3) compute the odd girth of G_T^d. The first step can be done in $\mathrm{O}(d|T|+d^2)$ time, the second and the third in $\mathrm{O}(d^2)$ time. Therefore, the inequality $\mathrm{sp}_T(C_{2n+1})\leqslant d-1$ can be tested in $\mathrm{O}(d|T|+d^2)$ time. With the aid of binary search and the following inequality:

$$\mathrm{sp}_T(G)\leqslant 2|T|,$$

which holds for all tripartite graphs (see [9,10] for far more general shape of the inequality), we are able to find the exact value of $\mathrm{sp}_T(C_{2n+1})$. The algorithm performs at most $\mathrm{O}(\log|T|)$ steps; the complexity of each step is $\mathrm{O}(|T|^2)$ since all values of d that are used satisfy the inequality $d\leqslant 2|T|$. Thus, the computational complexity of the algorithm is $\mathrm{O}(|T|^2\log|T|)$.

Claim 19. *For every two pairs u_1,v_1 and u_2,v_2 of adjacent vertices of a cycle, there is an automorphism φ of the cycle such that $\varphi(u_1)=u_2$ and $\varphi(v_1)=v_2$.*

Theorem 20. *If G is any non-bipartite thorny graph then $G\to C_{\mathrm{og}(G)}$.*

Proof. Let $G_1,G_2,\ldots,G_k$ be a CP-decomposition of G. We will use induction on i to show that if $i\in\{1,2,\ldots,k\}$ and $H_i=G_1\cup G_2\cup\cdots\cup G_i$ then $H_i\to C_{\mathrm{og}(G)}$.

The claim is true for $i=1$ since G_1 is a bipartite graph or an odd cycle satisfying $n(G_1) \geqslant \mathrm{og}(G)$. Assume that the claim holds for i ($i<k$) and let us check whether it holds for $i+1$. Since G_{i+1} is a bipartite graph or an odd cycle with $n(G_{i+1}) \geqslant \mathrm{og}(G)$, we see that $G_{i+1} \to C_{\mathrm{og}(G)}$. Let $h_1 : V(H_i) \to V(C_{\mathrm{og}(G)})$ and $h_2 : V(G_{i+1}) \to V(C_{\mathrm{og}(G)})$ be homomorphisms. By Claim 19, there is an automorphism $\varphi : V(C_{\mathrm{og}(G)}) \to V(C_{\mathrm{og}(G)})$ such that $\varphi(h_2(v)) = h_1(v)$ for each $v \in V(H_i) \cap V(G_{i+1})$. It is easy to verify that function $h : V(H_{i+1}) \to V(C_{\mathrm{og}(G)})$ defined as follows:

$$h(w) = \begin{cases} h_1(w) & \text{if } w \in V(H_i), \\ \varphi(h_2(w)) & \text{if } w \in V(G_{i+1}) \end{cases}$$

is a homomorphism. □

Now we are ready to show how to compute the T-span of any thorny graph G and how to find an optimal T-coloring of G, i.e. a T-coloring with span equal to $\mathrm{sp}_T(G)$. We start with the following observation—the method used to prove the above theorem may also serve as a description of an algorithm for finding a homomorphism from any non-bipartite thorny graph G to $C_{\mathrm{og}(G)}$. Previous considerations shows that the algorithm runs in time $\mathrm{O}(n^3)$ (the cost of finding a CP-decomposition is $\mathrm{O}(n^3)$; the algorithm performs also $k \leqslant n$ steps which are realizable in $\mathrm{O}(n^2)$ time).

Corollary 21. *If G is a thorny graph then*

$$\mathrm{sp}_T(G) = \begin{cases} 0 & \textit{if } G \textit{ is edgeless}, \\ m_T & \textit{if } G \textit{ is bipartite but not edgeless}, \\ \mathrm{sp}_T(C_{\mathrm{og}(G)}) & \textit{otherwise}. \end{cases}$$

Theorem 22. *The T-span of any thorny graph G can be found in time $\mathrm{O}(n^2 + |T|^2 \log |T|)$. Moreover, it is possible to find an optimal T-coloring of G in time $\mathrm{O}(n^3 + |T|^2 \log |T|)$.*

Proof. By Corollary 21, to compute the T-span of a thorny graph G it suffices to make at most two tests (check whether G is edgeless/bipartite) and select the proper number from 0, $\min \bar{T}$, $\mathrm{sp}_T(C_{\mathrm{og}(G)})$ as the T-span depending on these tests. Since both these tests can be done in $\mathrm{O}(n^2)$ time, the odd girth of G can be found in time $\mathrm{O}(n^2)$ (by Proposition 6, $m = \Theta(n)$) and the complexity of computing $\mathrm{sp}_T(C_{\mathrm{og}(G)})$ is $\mathrm{O}(|T|^2 \log |T|)$, we see that the computational complexity of the algorithm is $\mathrm{O}(n^2 + |T|^2 \log |T|)$.

Finding an optimal T-coloring of G is not far more complicated. It can be easily done in $\mathrm{O}(n^2 + |T|)$ time if G is bipartite. If G is not bipartite, it suffices to find a CP-decomposition of G (realizable in $\mathrm{O}(n^3)$ time), compute the odd girth of G (realizable in $\mathrm{O}(n^2)$ time), use the method described in the proof of Theorem 20 to find a homomorphism from G to $C_{\mathrm{og}(G)}$ (realizable in $\mathrm{O}(n^3)$ time) and find an optimal T-coloring of $C_{\mathrm{og}(G)}$ (realizable in $\mathrm{O}(|T|^2 \log |T|)$ time). Thus, the overall complexity is $\mathrm{O}(n^3 + |T|^2 \log |T|)$ time. □

Summarizing, we have just proved that the problem of finding the T-span and optimal T-colorings of thorny graphs is polynomially solvable. Our algorithms can be applied to solve the problem of finding the T-span for a wider class of graphs, namely bipartite and thus non-bipartite for which $G \to C_{\mathrm{og}(G)}$. Unfortunately, it is known [2] that it is NP-complete to verify whether $G \to C_{\mathrm{og}(G)}$ even if G is a subcubic graph.

References

[1] M.B. Cozzens, F.S. Roberts, T-colorings of graphs and the channel assignment problem, Congr. Numer. 35 (1982) 191–208.

[2] K. Giaro, R. Janczewski, M. Małafiejski, The complexity of T-coloring problem for graphs with small degree, Discrete Appl. Math., [doi: 10.1016/S0166-218X(02)00576-0].

[3] A. Gräf, Distance graphs and the T-coloring problem, Discrete Math. 196 (1999) 153–166.

[4] W.K. Hale, Frequency assignment: theory and applications, Proc. IEEE 68 (1980) 1497–1514.

[5] K. Jansen, A rainbow about T-colorings for complete graphs, Discrete Math. 154 (1996) 129–139.

[6] D.D.-F. Liu, T-colorings of graphs, Discrete Math. 101 (1992) 203–212.

[7] A. Raychaudhuri, Further results on T-colorings and frequency assignment problem, Discrete Math. 7 (1994) 605–613.

[8] F.S. Roberts, T-colorings of graphs: recent results and open problems, Discrete Math. 93 (1991) 229–245.

[9] B. Tesman, T-colorings, list T-colorings and set T-colorings of graphs, Ph.D. Thesis, Department of Mathematics, Rutgers University, New Brunswick, NJ, 1989.

[10] B. Tesman, Applications of forbidden difference graphs to T-colorings, Congr. Numer. 74 (1990) 15–24.