Discussiones Mathematicae
Graph Theory 35 (2015) 427–437
doi:10.7151/dmgt.1806

COMPLETELY INDEPENDENT SPANNING TREES IN (PARTIAL) k-TREES

Masayoshi Matsushita[1], Yota Otachi[2]

and

Toru Araki[1]

[1] *Department of Computer Science*
Gunma University, Kiryu
Gunma 376-8515 Japan

[2] *School of Information Science*
Japan Advanced Institute of Science and Technology
Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan

e-mail: otachi@jaist.ac.jp
arakit@gunma-u.ac.jp

Abstract

Two spanning trees T_1 and T_2 of a graph G are completely independent if, for any two vertices u and v, the paths from u to v in T_1 and T_2 are internally disjoint. For a graph G, we denote the maximum number of pairwise completely independent spanning trees by $\mathrm{cist}(G)$. In this paper, we consider $\mathrm{cist}(G)$ when G is a partial k-tree.

First we show that $\lceil k/2 \rceil \leq \mathrm{cist}(G) \leq k-1$ for any k-tree G. Then we show that for any $p \in \{\lceil k/2 \rceil, \ldots, k-1\}$, there exist infinitely many k-trees G such that $\mathrm{cist}(G) = p$. Finally we consider algorithmic aspects for computing $\mathrm{cist}(G)$. Using Courcelle's theorem, we show that there is a linear-time algorithm that computes $\mathrm{cist}(G)$ for a partial k-tree, where k is a fixed constant.

Keywords: completely independent spanning trees, partial k-trees.

2010 Mathematics Subject Classification: 05C05, 05C38.

1. Introduction

Let G be a simple undirected graph. The vertex set and the edge set of G are denoted by $V(G)$ and $E(G)$, respectively. The *degree* of a vertex v is denoted by $\deg_G v$. For a vertex x, $G - x$ is the graph obtained from G by removing x and all edges incident to x. A *clique* is a set of vertices that induces a complete subgraph. We call a clique of size k a *k-clique*. Let P_1 and P_2 be paths from a vertex x to a vertex y. If P_1 and P_2 are edge-disjoint and have no common vertex except for x and y, then the two paths are *internally disjoint*. A *spanning tree* of a connected graph G is a tree that is a subgraph of G and contains all vertices of G. Two spanning trees T_1 and T_2 of G are *completely independent* if, for any two distinct vertices u and v of G, the two u–v paths on T_1 and T_2 are internally disjoint.

The concept of completely independent spanning trees was introduced by Hasunuma [7] as a non-rooted variant of *independent spanning trees*. These concepts are related to fault-tolerant communication in interconnection networks. In [7], a characterization of completely independent spanning trees was shown, and it was also shown that there exist k completely independent spanning trees in the underlying graph of any k-connected line digraph. In [8], it was shown that there are two completely independent spanning trees in any 4-connected maximal planar graph, and a linear-time algorithm for finding such trees was proposed. Recently, in [9], it was shown that there are two completely independent spanning trees in the Cartesian product of two 2-connected graphs. In [11], Péterfalvi showed that, for any k, there is a k-connected graph that does not have two completely independent spanning trees. In [1], Araki showed that some sufficient conditions for Hamiltonian graphs are also sufficient conditions for graphs with two completely independent spanning trees.

In this paper, we study completely independent spanning trees of partial k-trees. The concept of partial k-trees was introduced as a generalization of trees. For $k \geq 1$, a *k-tree* is a graph constructed as follows.

1. A complete graph on k vertices is a k-tree.
2. If G is a k-tree and C is a k-clique of G and $x \notin V(G)$, then the new graph G' with the vertex set $V(G') = V(G) \cup \{x\}$ and the edge set $E(G') = E(G) \cup \{xc \mid c \in C\}$ is a k-tree.

A graph is a *partial k-tree* if it is a subgraph of a k-tree. For example, 1-trees are trees, and some 4-trees with small number of vertices are given in Figure 1. The partial k-trees are also known as the graphs of *treewidth* at most k (see e.g. [3]). Many algorithmic problems have been proved to be solvable in polynomial time for partial k-trees, if the parameter k is a fixed constant [5, 2]. Such algorithms are designed by using dynamic programming approach or description methods

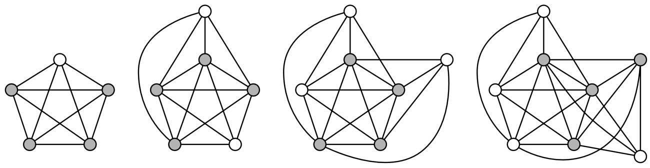

Figure 1. Examples of 4-trees. Gray vertices form 4-cliques.

from mathematical logic, in particular, the monadic second-order (MSO) logic. Courcelle [6] showed that every problem expressible in MSO can be solved in linear time on partial k-trees.

We discuss the number of completely independent spanning trees in a k-tree in Section 2. In Section 3, we show that the property of completely independent spanning trees can be expressed in MSO. This means that there is a linear-time algorithm for deciding whether there exist t completely independent spanning trees in a given partial k-tree.

2. Completely Independent Spanning Trees in k-trees

In this section, we consider the number of completely independent spanning trees in a k-tree. We will use the following characterization.

Theorem 2.1 (Hasunuma [7])**.** *Spanning trees $T_1, T_2, \ldots, T_t$ in G are pairwise completely independent if and only if they are edge-disjoint, and for any vertex v, there is at most one T_i such that $\deg_{T_i} v > 1$.*

The characterization says that every vertex of G is a non-leaf of at most one tree T_i. In other words, vertices of G can be colored with t colors in such a way that if a vertex is a non-leaf of T_i, then its color is i. A vertex that is a leaf of every tree can be colored arbitrarily.

For a graph G, we denote the maximum number of pairwise completely independent spanning trees by $\mathrm{cist}(G)$. We consider $\mathrm{cist}(G)$ when G is a k-tree.

We first show that $\mathrm{cist}(K_n) = \lfloor n/2 \rfloor$ for $n \geq 4$. Note that Hasunuma [8] mentioned this fact for even n without proof.

Lemma 2.2. *For $n \geq 4$, $\mathrm{cist}(K_n) = \lfloor n/2 \rfloor$.*

Proof. Since K_n has $n(n-1)/2$ edges and completely independent spanning trees are edge-disjoint by Theorem 2.1, we obtain

$$\mathrm{cist}(K_n) \leq \left\lfloor \frac{n(n-1)/2}{n-1} \right\rfloor = \lfloor n/2 \rfloor.$$

Let $V(K_n) = \{v_1, v_2, \dots, v_n\}$. First assume that $n = 2t$. For $1 \le i \le t$, let $V_i = \{v_{2i-1}, v_{2i}\}$. We define a spanning tree T_i, $1 \le i \le t$, as follows.

1. Edge $v_{2i-1}v_{2i}$ is an edge of T_i.
2. For $1 \le j < i$, edges $v_{2i-1}v_{2j}$ and $v_{2i}v_{2j-1}$ are edges of T_i.
3. For $i < j \le t$, edges $v_{2i-1}v_{2j-1}$ and $v_{2i}v_{2j}$ are edges of T_i.

We can easily check for any $j \ne i$ that two spanning trees T_i and T_j have no common edge, and the vertices in V_i have degree 1 in T_j. Hence, by Theorem 2.1, these t spanning trees are completely independent.

Next we assume that $n = 2k + 1$. In this case, first we construct t completely independent spanning trees $T_1, T_2, \dots, T_t$ in K_{2t}. Now let v_{2t+1} be the vertex of K_{2t+1} but not in K_{2t}. By adding a vertex v_{2t+1} and an edge $v_{2t+1}v_{2i-1}$ to T_i, we construct the spanning tree T'_i of K_{2t+1}. Obviously, $T'_1, \dots, T'_t$ are completely independent. ∎

As mentioned earlier, if G has t completely independent spanning trees $T_1, \dots, T_t$, then the vertices of G can be colored with t colors in such a way that if a vertex is a non-leaf of T_i, then its color is i. By the proof of Lemma 2.2, the vertices of K_n can be colored with $\lfloor n/2 \rfloor$ colors such that the two vertices in V_i have color i. If n is odd, then the vertex v_n can be colored with any color. Examples of the construction of completely independent spanning trees in K_6 and K_7 are in Figures 2 and 3, respectively. The numbers in the vertices are the assigned colors .

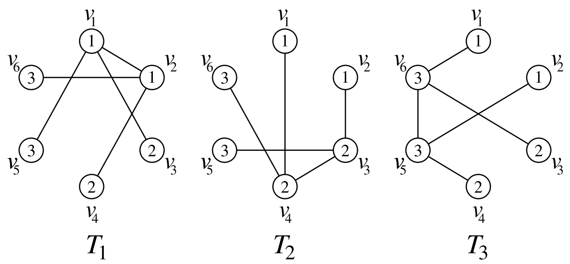

Figure 2. Three completely independent spanning trees in K_6.

Theorem 2.3. *For $k \ge 3$ and a k-tree G on at least $k + 1$ vertices,* $\operatorname{cist}(G) \ge \lceil k/2 \rceil$.

Proof. Let $t = \lceil k/2 \rceil$. First we assume that k is odd. We show the theorem by induction on the number of vertices.

When G has $k + 1$ vertices, it is a complete graph on $k + 1$ vertices. Hence by Lemma 2.2 it has $\lfloor (k + 1)/2 \rfloor = t$ completely independent spanning trees

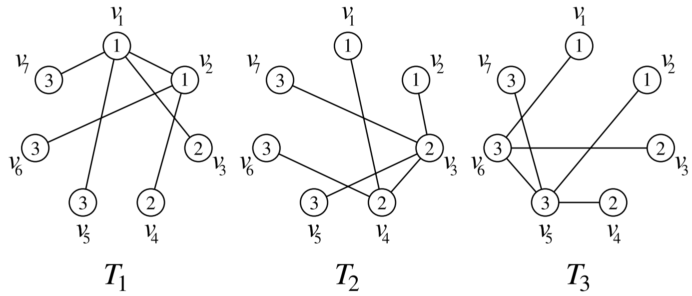

Figure 3. Three completely independent spanning trees in K_7.

$T_1, T_2, \ldots, T_t$. Furthermore, the vertices can be colored with t colors such that the two inner vertices of T_i has color i.

Now let $n \geq k+1$ and assume that any k-tree G on n vertices has t completely independent spanning trees $T_1, T_2, \ldots, T_t$, and the vertices in G are colored with t colors such that the inner vertices of T_i has color i, and for any $(k+1)$-clique C, there are t colors in C and there are two vertices of each color. Observe that any k-clique is included in a $(k+1)$-clique in a connected k-tree with at least $k+1$ vertices. Thus any k-clique has vertices of t colors and there are two vertices of each color except for one color. For example, for a complete graph in Figure 2, in a clique induced by five vertices v_i, $1 \leq i \leq 5$, there are two vertices of color 1 and 2, and one vertex of color 3 ($k = 5$ and $t = 3$ in this case).

Let H be a k-tree on $n+1$ vertices. Since H is a k-tree, there is a vertex v of degree k such that the neighborhood of v induces a clique C of size k, and $G := H - v$ is a k-tree on the n vertices. By induction hypothesis, G has t completely independent spanning trees. Suppose that $u_1, u_2, \ldots, u_t$ are vertices of C such that u_i has color i and no vertex other than u_t has color t in C. A spanning tree T_i', $i = 1, 2, \ldots, t$, of H is constructed by adding a vertex v and edges vu_i to T_i. The spanning trees T_i', $i = 1, 2, \ldots, t$, are completely independent by Theorem 2.1.

Since the vertex v is a leaf of every spanning tree, we can assign any color to v. Hence we color the vertex v with color t. Then, in the clique induced by $k+1$ vertices $C \cup \{v\}$, there are two vertices of each color. Hence, colors in any clique of size $k+1$ of H satisfy the induction hypothesis.

For even $k = 2t$, we can prove similarly by changing the induction hypothesis as "there are t completely independent spanning trees $T_1, \ldots, T_t$ in G, and the vertices are colored with t colors such that the inner vertices of T_i has color i, and for any clique C of size $k+1$, there are t colors in C and there are at least two vertices of each color." ■

Examples of the construction of two completely independent spanning trees

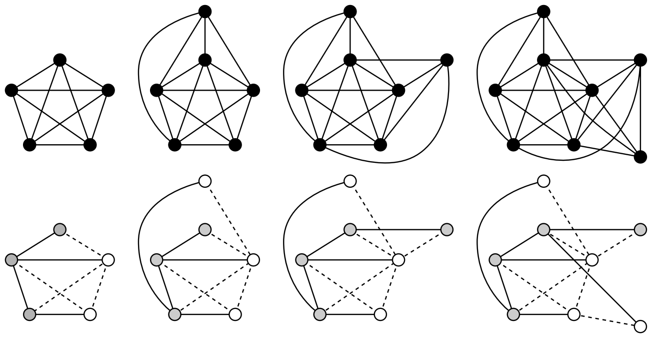

Figure 4. Two completely independent spanning trees in 4-trees. The vertices of colors 1 and 2 are drawn by gray and white vertices, respectively. The edges of two spanning trees T_1 and T_2 are solid lines and dotted lines, respectively.

in 4-trees are illustrated in Figure 4. In Figure 4, the vertices of colors 1 and 2 are drawn by gray and white vertices, respectively. By assigning appropriate colors to the newly added vertex, we can keep the condition that each 5-clique has two colors and at least two vertices of each color.

The following upper bound can be easily derived by considering the number of edges in a k-tree.

Lemma 2.4. *For $k \geq 2$ and a k-tree G, $\mathrm{cist}(G) \leq k - 1$.*

Proof. From the definition of k-trees, we can see that a k-tree on n vertices has $kn - k(k+1)/2$ edges: the first k vertices form a k-clique, and each other vertex adds k edges. Since a spanning tree has $n - 1$ edges, it follows that

$$\mathrm{cist}(G) \leq \left\lfloor \frac{kn - k(k+1)/2}{n-1} \right\rfloor .$$

Since the right-hand side is at most $k - 1$ for $k \geq 2$, the lemma holds. ■

Now Theorem 2.3 and Lemma 2.4 together imply the following fact.

Corollary 2.5. *If $k \geq 3$, then $\lfloor (k+1)/2 \rfloor \leq \mathrm{cist}(G) \leq k - 1$ for any k-tree G.*

In order to show the next theorem, we define a class of k-trees $\mathcal{G}(k, p)$ for $p < k$. Let $k \geq 3$, $p \geq 2$ and $n \geq (p^2 + p + 2)/2$. A graph $G \in \mathcal{G}(k, p)$ on n vertices has $V \cup U \cup W$ as its vertex set, where $V = \{v_1, \ldots, v_{2p}\}$ and $U = U_1 \cup \cdots \cup U_{p-2}$, $U_j = \{u_{j,j}, u_{j,j+1}, \ldots, u_{j,p-2}\}$, and $W = \{w_1, \ldots, w_{n-(p^2+p+2)/2}\}$. The edges of G are as follows.

- The vertices $v_{i+1}, v_{i+2}, \ldots, v_{i+k}$ form a k-clique for $0 \leq i \leq 2p - k$.

- For $1 \le j \le p-2$, each vertex in U_j is adjacent to all vertices in a k-clique C such that $\{v_j, v_{j+1}, \dots, v_{j+p}\} \subseteq C \subseteq V$.
- Each vertex in W is adjacent to the k vertices $v_1, v_2, \dots, v_k$.

We can see that G is a k-tree. The class $\mathcal{G}(k,p)$ consists of all such k-trees.

Theorem 2.6. *For $k \ge 3$ and any $\lfloor (k+1)/2 \rfloor \le p \le k-1$, and for $n \ge (p^2+p+2)/2$, there is a k-tree G on n vertices such that* $\mathrm{cist}(G) = p$.

Proof. We show that $\mathrm{cist}(G) = p$ for any $G \in \mathcal{G}(k,p)$. For $1 \le i \le p$, a spanning tree T_i is defined as follows.

- Vertex v_i is adjacent to the vertices in $\{v_{i+1}, v_{i+2}, \dots, v_{i+p}\}$, and v_{i+p} is adjacent to the vertices in $\{v_{i-1}\} \cup \{v_{i+p+1}, v_{i+p+2}, \dots, v_{2p}\}$.
- Vertex v_i is adjacent to the vertices in $U_1 \cup U_2 \cup \cdots \cup U_{i-1}$, and v_{i+p} is adjacent to the vertices in $U_i \cup U_{i+1} \cup \cdots \cup U_{p-2}$.
- When $i \ge 3$, v_j is adjacent to $u_{j,i-2}$ for $1 \le j \le i-2$.
- Vertex v_i is adjacent to the vertices in W.

The resulting graph is T_i.

We show that $T_1, T_2, \dots, T_p$ are completely independent. We can see that these spanning trees have no common edge. For T_i, the vertices of degree at least 2 are v_i, v_{i+p} and $u_{1,i-2}, u_{2,i-2}, \dots, u_{i-2,i-2}$. Hence, there is no vertex x such that $\deg_{T_i} x \ge 2$ and $\deg_{T_j} x \ge 2$ if $i \ne j$. Thus, by Theorem 2.1, these p spanning trees are completely independent. Hence G has p completely independent spanning trees.

Next we show that G cannot have $p+1$ completely independent spanning trees. Suppose that G has $p+1$ pairwise completely independent spanning trees $T_1, T_2, \dots, T_{p+1}$. Then we assign $p+1$ colors to the vertices of G so that a non-leaf of T_i has color i. Consider the colors of the vertices $V = \{v_1, v_2, \dots, v_{2p}\}$. Each tree T_i has at least two inner vertices and the inner vertices induce a connected graph. Since $U \cup W$ is an independent set, every color appears at least once in V. Since V has $2p$ vertices and $p+1$ colors, there are two colors such that V contains exactly one vertex for each of the colors. Without loss of generality, assume that 1 and 2 are such colors, $v_i \in V$ has color 1, and $v_j \in V$ has color 2. Now the vertices in $V \setminus \{v_i\}$ have degree 1 in the spanning tree T_1, and the vertices in $V \setminus \{v_j\}$ have degree 1 in the spanning tree T_2. Since $U \cup W$ is an independent set, every edge between v_i and $U \cup W$ is an edge of T_1. Similarly, every edge between v_j and $U \cup W$ is an edge of T_2. Since v_j is a leaf of T_1, edge $v_i v_j$ is included in T_1. Similarly, edge $v_i v_j$ must be in T_2. This contradicts that T_1 and T_2 are edge-disjoint. ∎

For example, five completely independent spanning trees of a k-tree $G \in \mathcal{G}(6,5)$ on 16 vertices are presented in Figure 5.

3. Complexity for Partial k-trees

In this section, we consider algorithmic aspects for constructing completely independent spanning trees. Here we use a very strong and general tool for considering time complexity of problems on partial k-trees.

The syntax of the *monadic second-order* (MSO) logic of graphs includes the logical connectives $\wedge$, $\vee$, $\neg$, $\Longrightarrow$, and $\Longleftrightarrow$, variables for vertices, edges, sets of vertices, and sets of edges, the quantifiers $\forall$ and $\exists$ that can be applied to these variables, and the following binary relations.

- $\mathrm{inc}(e,v)$, where e is an edge variable, v is a vertex variable, and the interpretation is that the edge e is incident to the vertex v.
- $x \in X$, where x is a vertex (an edge) variable and X is a vertex set (an edge set, respectively) variable.
- Equality, $=$, of variables representing vertices, edges, set of vertices and set of edges.

It is known that several problems on graphs can be expressed in MSO. See [10] for more details. Courcelle [6] provided the following theorem that gives a general framework for efficient algorithms on partial k-trees.

Theorem 3.1 (Courcelle [6]). *Every problem expressible in* MSO *can be solved in linear time on partial k-trees for any fixed k.*

We now prove that the problem of finding t completely independent spanning trees is expressible in MSO.

Lemma 3.2. *The property that a graph has t completely independent spanning trees can be expressed in* MSO.

Proof. For $F \subseteq E(G)$, we denote by $G[F]$ the subgraph induced by F. It is easy to see that the following properties can be expressed in MSO (see [4]).

- $\mathrm{SpnTree}(E_1)$: $G[E_1]$ is a spanning tree of G.
- $\mathrm{Deg1}(v_1, E_1)$: v_1 has at most one neighbor in $G[E_1]$.

Now the following expression means $E_1, E_2, \ldots, E_t$ induce edge disjoint spanning trees.

$$\mathrm{EDST}(E_1,\ldots,E_t) := \bigwedge\nolimits_{1\le i\le t} \mathrm{SpnTree}(E_i) \wedge \bigwedge\nolimits_{1\le i<j\le t} ((\forall e_1)(e_1 \notin E_i \vee e_1 \notin E_j)).$$

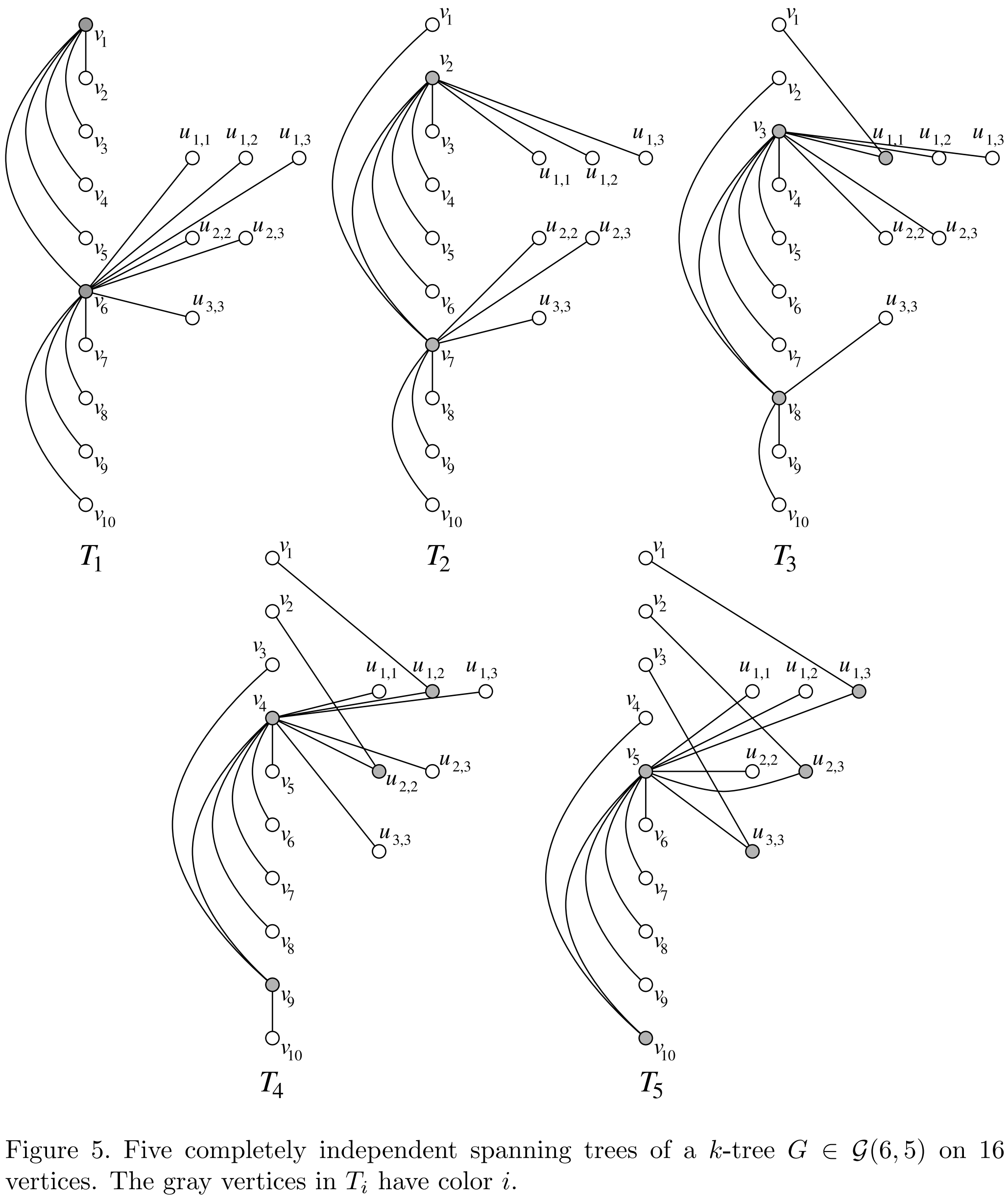

Figure 5. Five completely independent spanning trees of a k-tree $G \in \mathcal{G}(6, 5)$ on 16 vertices. The gray vertices in T_i have color i.

Next we define the expression meaning that for each vertex v_1 there is at most one edge set E_i such that v_1 is not a vertex of degree at most 1 in $G[E_i]$.

$$\mathrm{Uniq}(E_1, \ldots, E_t) := (\forall v_1)(\exists E_i)(\forall E_j)\big(E_j \neq E_i \implies \mathrm{Deg1}(v_1, E_j)\big).$$

By Theorem 2.1, the graph G has t completely independent spanning trees if and only if

$$G \models (\exists E_1, \ldots, E_t)(\mathrm{EDST}(E_1, \ldots, E_t) \wedge \mathrm{Uniq}(E_1, \ldots, E_t)).$$

The lemma follows. ∎

By Theorem 3.1 and Lemma 3.2, it follows that given a partial k-tree G, the problem of deciding whether G has at least t completely independent spanning trees can be solved in linear time for any fixed t and k. Observe that $\mathrm{cist}(G) \leq k-1$ from the same argument we used for k-trees in the proof of Lemma 2.4. Hence, to determine $\mathrm{cist}(G)$, it suffices to test for each $t \in \{1, \ldots, k-1\}$ whether $\mathrm{cist}(G) \leq t$. If k is a fixed constant, then the number of tests is a constant $k-1$, and each test can be done in linear time since $t < k$ is also a constant. Thus the following theorem holds.

Theorem 3.3. *Given a partial k-tree G, where k is a fixed constant,* $\mathrm{cist}(G)$ *can be determined in linear time.*

Acknowledgment

This study was supported in part by JSPS KAKENHI Grant Numbers 23500007, 25730003.

References

[1] T. Araki, *Dirac's condition for completely independent spanning trees*, J. Graph Theory **77** (2014) 171–179.
doi:10.1002/jgt.21780

[2] S. Arnborg and A. Proskurowski, *Linear time algorithms for NP-hard problems restricted to partial k-trees*, Discrete Appl. Math. **23** (1989) 11–24.
doi:10.1016/0166-218X(89)90031-0

[3] H.L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci. **209** (1998) 1–45.
doi:10.1016/S0304-3975(97)00228-4

[4] H.L. Bodlaender, F.V. Fomin, P.A. Golovach, Y. Otachi and E.J. van Leeuwen, *Parameterized complexity of the spanning tree congestion problem*, Algorithmica **64** (2012) 85–111.
doi:10.1007/s00453-011-9565-7

[5] H.L. Bodlaender and A.M.C.A. Koster, *Combinatorial optimization on graphs of bounded treewidth*, Comput. J. **51** (2008) 255–269.
doi:10.1093/comjnl/bxm037

[6] B. Courcelle, *The monadic second-order logic of graphs III: tree-decompositions, minors and complexity issues*, Theor. Inform. Appl. **26** (1992) 257–286.

[7] T. Hasunuma, *Completely independent spanning trees in the underlying graph of a line digraph*, Discrete Math. **234** (2001) 149–157.
doi:10.1016/S0012-365X(00)00377-0

[8] T. Hasunuma, *Completely independent spanning trees in maximal planar graphs* in: Proceedings of 28th Graph Theoretic Concepts of Computer Science (WG 2002), LNCS **2573**, Springer-Verlag Berlin (2002) 235–245.
doi:10.1007/3-540-36379-3_21

[9] T. Hasunuma and C. Morisaka, *Completely independent spanning trees in torus networks*, Networks **60** (2012) 59–69.
doi:10.1002/net.20460

[10] P. Hliněný, S. Oum, D. Seese and G. Gottlob, *Width parameters beyond tree-width and their applications*, Comput. J. **51** (2008) 326–362.
doi:10.1093/comjnl/bxm052

[11] F. Péterfalvi, *Two counterexamples on completely independent spanning trees*, Discrete Math. **312** (2012) 808–810.
doi:10.1016/j.disc.2011.11.015

Received 24 April 2014
Revised 8 August 2014
Accepted 8 August 2014