Connecting Vertices by Independent Trees*

Manu Basavaraju[1], Fedor V. Fomin[1], Petr A. Golovach[1], and Saket Saurabh[1,2]

1 **Department of Informatics, University of Bergen, PB 7803, 5020 Bergen, Norway**
{manu.basavaraju,fedor.fomin,petr.golovach}@ii.uib.no
2 **Institute of Mathematical Sciences, Chennai, India**
saket@imsc.res.in

Abstract

We study the paramereteized complexity of the following connectivity problem. For a vertex subset U of a graph G, trees $T_1, \ldots, T_s$ of G are completely independent spanning trees of U if each of them contains U, and for every two distinct vertices $u, v \in U$, the paths from u to v in $T_1, \ldots, T_s$ are pairwise vertex disjoint except for end-vertices u and v. Then for a given $s \geq 2$ and a parameter k, the task is to decide if a given n-vertex graph G contains a set U of size at least k such that there are s completely independent spanning trees of U. The problem is known to be NP-complete already for $s = 2$. We prove the following results:

- For $s = 2$ the problem is solvable in time $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$.
- For $s = 2$ the problem does not admit a polynomial kernel unless NP $\subseteq$ coNP /poly.
- For arbitrary s, we show that the problem is solvable in time $f(s, k)n^{\mathcal{O}(1)}$ for some function f of s and k only.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases Parameterized complexity, FPT-algorithms, completely independent spanning trees

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2014.73

1 Introduction

Two spanning trees T_1 and T_2 of a graph G are *independent* if they are rooted in the same vertex r and for every vertex $v \neq r$ of G, the two (v, r)-paths, one in T_1 and one in T_2, are internally disjoint, i. e. having no edge and no internal vertex in common. Independent spanning trees have applications to fault-tolerant protocols in distributed processor networks [3, 11]. In 2001, Hasunuma in [7, 8] introduced the notion of *completely independent spanning trees*, an interesting variant of the classical notion of connectivity. Formally, spanning trees $T_1, \ldots, T_s$ of a graph G are *completely independent* if for every two distinct vertices $u, v \in V(G)$, the (u, v)-paths in $T_1, \ldots, T_s$, are pairwise vertex disjoint except for end-vertices u and v.

The problem of deciding whether a graph G has two completely independent spanning trees is NP-complete [8]. Since not every graph has even two completely independent spanning trees, the following optimization version of the problem is meaningful. For a given

* Supported by the European Research Council (ERC) via grant Rigorous Theory of Preprocessing, reference 267959.

34th Int'l Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2014).
Editors: Venkatesh Raman and S. P. Suresh; pp. 73–84
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$s \geq 2$, can one find a maximum set of vertices spanned by s completely independent trees? More precisely, for a set of vertices U of a graph G, we say that a subgraph T of G is a *spanning tree of U* if T is an inclusion-minimal tree in G containing all vertices of U. Spanning trees $T_1, \ldots, T_s$ of U are completely independent if for any two distinct vertices $u, v \in U$, the (u, v)-paths in $T_1, \ldots, T_s$, are pairwise vertex disjoint except for end-vertices u and v. Then the task is to find a set of vertices U of maximum size (we call the vertices of U *terminals*) such that there are s completely independent spanning trees of U.

In this paper, we initiate the study of the following parameterized problem.

Independently s-Connected k-Set	
Instance:	A graph G and positive integers $s \geq 2$ and k.
Parameter 1:	s.
Parameter 2:	k.
Question:	Does G contain a set of terminals U of size at least k such that there are s completely independent spanning trees of U?

Previous results. Hasunama [8] has shown that it is NP-complete to decide whether a graph G has two completely independent spanning trees. He also obtained a number of results about existence of completely independent spanning trees for some special graph classes. Other, mostly combinatorial, studies of the problem were carried out by Hasunuma and Morisaka [9] and Péterfalvi [12].

Our contribution. Our main result is stated in the following theorem.

▶ **Theorem 1.** Independently 2-Connected k-Set *can be solved in time* $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ *for n-vertex graphs.*

We prove the theorem by applying a WIN/WIN approach. We start with a combinatorial result, which is interesting on its own. In Section 3 we show that every 2-connected graph of pathwidth at least k, contains as a minor a graph H, which is a tree on k vertices plus one vertex adjacent to all other vertices. We also give a polynomial time algorithm which either provide us H, or a path decomposition of width $k-1$. As it is sufficient to solve Independently 2-Connected k-Set for the blocks of the input graph, we either obtain two completely independent spanning trees for k terminals, or construct a path decomposition of width at most $k-1$. The next step is an algorithm given in Section 5 that solves Independently 2-Connected k-Set in time single exponential in the treewidth of the input graph. This step is based on the recent techniques of computing representative sets of graphic matroids [4]. Combining together both cases, we obtain the proof of Theorem 1.

Let us remark, that the NP-hardness reduction in [8] from Not-All-Equal-3SAT reduces to a graph of size linear in the number of variables and clauses of the formula. Thus, unless the Exponential Time Hypothesis of Impagliazzo, Paturi, and Zane [10] fails, there is no $2^{o(k)} n^{\mathcal{O}(1)}$ algorithm for Independently 2-Connected k-Set and thus our upper bound is asymptotically tight up to ETH.

We complement our algorithm with a complexity result on kernelization for Independently 2-Connected k-Set, namely that the problem does not admit a polynomial kernel unless NP $\subseteq$ coNP /poly.

We also show that Independently s-Connected k-Set is FPT when parameterized by $s + k$. It is not hard to reduce Independently s-Connected k-Set to the problem of finding a topological minor of constant size in a graph. Then the result follows from a deep Theorem of Grohe, Kawarabayashi, Marx and Wollan [6] on the parameterized testing of topological minors.

2 Preliminaries

Graphs. We consider finite undirected graphs without loops or multiple edges. The vertex set of a graph G is denoted by $V(G)$ and the edge set is denoted by $E(G)$. For a set of vertices $S \subseteq V(G)$, $G[S]$ denotes the subgraph of G induced by S, and by $G - S$ we denote the graph obtained from G by the removal of all the vertices of S, i. e., the subgraph of G induced by $V(G) \setminus S$. For a single element set $\{v\}$, we write $G - v$ instead of $G - \{v\}$. For a vertex v, we denote by $N_G(v)$ its *(open) neighborhood* in G, that is, the set of vertices which are adjacent to v. The *degree* of a vertex v is denoted by $d_G(v) = |N_G(v)|$, and $\Delta(G)$ is the maximum degree of G. A vertex v is a *cut-vertex* of G if $G - v$ has more connected components than G. A connected graph with at least two vertices is 2-*connected* if it does not contain a cut-vertex. A maximal 2-connected subgraph of G is called a 2-*connected component* or *block* of G. Let T be a tree. For a vertex $v \in V(T)$, we say that v is a *leaf* if $d_T(v) = 1$ or $d_T(v) = 0$ (if $|V(T)| = 1$), and we say that v is an *internal* vertex otherwise.

Minors. The *edge contraction* of $e = uv$ removes u and v from G, and replaces them by a new vertex adjacent to precisely those vertices to which u or v were adjacent. If u is a vertex of degree two such that its neighbors x, y are not adjacent, then the *vertex dissolution* of u removes u and adds a new edge xy. A graph H is a *minor* of G if H can be obtained from a subgraph of G by a sequence of vertex deletions, edge deletions and edge contractions. Alternatively, we can define minors as follows. For two non-empty vertex disjoint subsets $X_1, X_2 \subseteq V(G)$, X_1 and X_2 are *adjacent* if there is $uv \in E(G)$ such that $u \in X_1$ and $v \in X_2$. An *H-witness structure* $\mathcal{W}$ is a collection of $|V(H)|$ non-empty vertex disjoint subsets $W(x) \subseteq V(G)$, one for each $x \in V(H)$, called *H-witness sets*, such that each $W(x)$ induces a connected subgraph of G, and for all $x, y \in V(H)$ with $x \neq y$, if x and y are adjacent in H, then $W(x)$ and $W(y)$ are adjacent in G. It is straightforward to see that H is a minor of G if and only if G has an H-witness structure. A graph H is a *topological minor* of G if H can be obtained from a subgraph of G by a sequence of vertex deletions, edge deletions and vertex dissolution. Notice that if H is a topological minor of G, then by subdividing edges of H we can obtain a graph that is isomorphic to a subgraph of G.

Treewidth and pathwidth. A *tree decomposition* of a graph G is a pair (X, T) where T is a tree and $X = \{X_i \mid i \in V(T)\}$ is a collection of subsets (called *bags*) of $V(G)$ such that:
1. $\bigcup_{i \in V(T)} X_i = V(G)$,
2. for each edge $xy \in E(G)$, $x, y \in X_i$ for some $i \in V(T)$, and
3. for each $x \in V(G)$, the set $\{i \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of a tree decomposition $(\{X_i \mid i \in V(T)\}, T)$ is $\max_{i \in V(T)} \{|X_i| - 1\}$. The *treewidth* of a graph G (denoted as $\mathbf{tw}(G)$) is the minimum width over all tree decompositions of G.

If T is restricted to be a path, then (X, T) is said to be a *path decomposition*. Respectively, the *pathwidth* of a graph G (denoted as $\mathbf{pw}(G)$) is the minimum width over all path decompositions of G. Whenever we consider a path decomposition (X, P), we assume that the bags are enumerated in the path order with respect to P. In other words, a path decomposition of G is a sequence of bags $(X_1, \dots, X_r)$.

3 Algorithm for Independently 2-Connected k-Set

In this section we design an algorithm for Independently 2-Connected k-Set. We start by a simple characterization of completely independent spanning trees that we use in our arguments. This is followed by a a structural result that shows that if the pathwidth of the input graph is large then the given instance is a YES instance. We use this to design a algorithm mentioned in Theorem 1.

3.1 Characterization of completely independent spanning trees

Hasunuma proved in [7] that if $T_1, \ldots, T_s$ are spanning trees of a graph G, then $T_1, \ldots, T_s$ are completely independent if and only if $T_1, \ldots, T_s$ are edge-disjoint and for any vertex $v \in V(G)$, there is at most one spanning tree T_i such that $d_{T_i}(v) > 1$. We need a similar claim for completely independent spanning trees of a set of terminals.

▶ **Lemma 2.** *Let G be a graph, and let $U \subseteq V(G)$ with $|U| = k$. Let also $T_1, \ldots, T_s$ be spanning trees of U. Then $T_1, \ldots, T_s$ are completely independent spanning trees of U if and only if*

1. *$T_1, \ldots, T_s$ are edge disjoint,*
2. *for all $i, j \in \{1, \ldots, s\}$, $i \neq j$, if $v \in V(T_i) \cap V(T_j)$, then $v \in U$,*
3. *for each $v \in U$, there is at most one $i \in \{1, \ldots, s\}$ such that $d_{T_i}(v) > 1$.*

Proof. We assume that $k, s \geq 2$, as the claim is trivial otherwise. We first show the forward direction. Suppose that $T_1, \ldots, T_s$ are completely independent spanning trees of U.

We show that for any $i, j \in \{1, \ldots, s\}$, $i \neq j$, T_i and T_j have no common vertex that is an internal vertex of both the trees. To obtain a contradiction, assume that $u \in V(T_i) \cap V(T_j)$ is an internal vertex of both T_i and T_j. The vertex u is a cut-vertex of T_i. Because T_i is an inclusion-minimal tree that contains U, there are two terminals $x, y \in U$ that are in two distinct components T_i' and T_i'' of $T_i - u$ respectively. The tree T_j has the unique (x, y)-path P and $u \notin V(P)$. Since u is an internal vertex of T_j, $T_j - u$ has at least two components, and P lies completely in one component T_j' of $T_j - u$. By minimality, there is $z \in U$ such that z is in another component of $T_j - u$. Notice that $z \notin V(T_i')$ or $z \notin V(T_i'')$. Assume without loss of generality that $z \notin V(T_i')$. Because $x \in V(P)$ and z are in distinct components of $T_j - u$, u is an internal vertex of the (x, z)-path in T_j. Because $z \notin V(T_i')$ and $x \in V(T_i')$, u is an internal vertex of the (x, z)-path in T_i as well, but it contradicts the assumption that $T_1, \ldots, T_s$ are completely independent spanning trees of U.

The proved claim immediately implies (3). To show (1), assume that two distinct trees T_i, T_j have a common edge uv. Because neither u nor v can be an internal vertex of the both trees, we can assume without loss of generality that u is a leaf of T_i and v is a leaf of T_j. Because T_i, T_j are inclusion-minimal trees that contains U, any leaf of T_i or T_j is a terminal, and $u, v \in U$. Then we have that the (u, v)-paths in T_i and T_j have a common edge; a contradiction. To prove (2), it is sufficient to observe that if $v \in V(T_i) \cap V(T_j)$ and $v \notin U$, then by minimality of T_i, T_j, v is an internal vertex of both these trees, a contradiction.

Assume now that $T_1, \ldots, T_s$ are spanning trees of U that satisfy (1)–(3). Consider any distinct $u, v \in U$ and $i, j \in \{1, \ldots, s\}$. Let P_i, P_j be the (u, v)-paths in T_i and T_j respectively. By (1), P_i and P_j are edge disjoint. If P_i and P_j have a common vertex $x \neq u, v$, then by (2), $x \in U$, and then $d_{T_i}(x), d_{T_j}(x) \geq 2$ contradicting (3). Hence, P_i and P_j are internally vertex disjoint. ◀

Clearly, if G is a disconnected subgraph, then G has a set of terminals U of size at least k such that there are s completely independent spanning trees of U if and only if there is such a set of terminals in one of the components of G, i. e., we can consider only connected graphs. Lemma 2 implies that we can restrict ourself by 2-connected graphs. To see it, it is sufficient to observe that if a set of terminals U has two vertices that does not belong to the same block, then there is a cut-vertex of G that is an internal vertex of any spanning tree of U contradicting Lemma 2.

▶ **Lemma 3.** *Let G be a connected graph. For positive integers s and k, G has a set of terminals U of size at least k such that there are s completely independent spanning trees of U in G if and only if there is a block H of G with the same property.*

3.2 Independent trees and pathwidth

In this section we show that if a 2-connected graph G has pathwidth at least k, then G has a set of terminals U of size at least k such that there are two completely independent spanning trees of U. We need some additional notations. Let G be a graph. For $Z \subseteq V(G)$, $\mathbf{att}(Z)$ is the set of all $v \in Z$ with a neighbor in $V(G) \setminus Z$, and $\alpha(Z) = |\mathbf{att}(Z)|$.

▶ **Theorem 4.** *Let G be a 2-connected graph with n vertices and m edges. Let also k be a positive integer. If $\mathbf{pw}(G) \geq k$, then G has a minor H with the property that there is a vertex $w \in V(H)$ such that $d_H(w) \geq k$ and $H - w$ is a tree. Moreover, there is an algorithm that in time $\mathcal{O}(nm)$ either produces a witness structure of such a minor H, or constructs a path decomposition of G of width at most $k-1$.*

Proof. Suppose that Z is a non-empty proper subset of $V(G)$ that satisfies the following conditions:

(i) $1 \leq \alpha(Z) \leq k$,

(ii) there are vertex disjoint connected subgraph $C_0, \ldots, C_t$ of $G[Z]$ where $t = \alpha(Z) - 1$ such that
- for each $i \in \{0, \ldots, t\}$, $V(C_i) \cap \mathbf{att}(Z) \neq \emptyset$,
- G has an edge with one end-vertex in C_0 and another in C_i for all $i \in \{1, \ldots, t\}$, and
- $V(C_1) \cup \ldots \cup V(C_t)$ are in the same component of $G - V(C_0)$.

(iii) $G[Z]$ has a path decomposition $(X_1, \ldots, X_r)$ of width at most $k-1$ such that $\mathbf{att}(Z) \subseteq X_r$.

Notice that $\mathbf{att}(Z) \subseteq V(C_0) \cup \ldots \cup V(C_t)$ and each C_i has the unique vertex in $\mathbf{att}(Z)$.

We prove the following claim.

▶ **Claim A.** *Either $\alpha(Z) = k$ and G has a minor H with the property that there is a vertex $w \in V(H)$ such that $d_H(w) \geq k$ and $H - w$ is a tree, or $|V(G) \setminus Z| = 1$ and $\mathbf{pw}(G) \leq k-1$, or there is Z' such that $Z \subset Z' \subset V(G)$ and Z' satisfies (i)–(iii).*

Proof of Claim A. Suppose that $\alpha(Z) = k = t+1$. Consider $u \in \mathbf{att}(Z) \cap V(C_0)$. There is a neighbor v of u in $V(G) \setminus Z$. Let C_{t+1} be the subgraph of G with the unique vertex v. The graph G is 2-connected. Then $G - u$ is connected, and G has a path that joins v with at least one of $C_1, \ldots, C_t$ that avoids C_0. Because $V(C_1) \cup \ldots \cup V(C_t)$ are in the same component of $G - V(C_0)$, we have that $V(C_1) \cup \ldots \cup V(C_{t+1})$ also are in the same component of $G - V(C_0)$. Now we construct the minor H of G as follows. We contract the edges of C_0 and denote the obtained vertex w. Then we contract the edges of the subgraphs $C_1, \ldots, C_k$ and denote the obtained vertices by $u_1, \ldots, u_k$ respectively. Let G' be the obtained graph. The vertices $u_1, \ldots, u_k$ are in the same component of $G' - w$. Hence, $G' - w$ has a tree T that contains $u_1, \ldots, u_k$. We remove the vertices of $V(G') \setminus (V(T) \cup \{w\})$. Finally, we remove all the edges of the obtained graph except the edges of T and the edges that join w and T. Because $u_1, \ldots, u_k \in V(T)$ are adjacent to w, we have a required minor.

Let now $\alpha(Z) < k$ and let $|V(G) \setminus Z| = 1$. By (iii), $G[Z]$ has a path decomposition $(X_1, \ldots, X_r)$ of width at most $k-1$ such that $\mathbf{att}(Z) \subseteq X_r$. Let $X_{r+1} = \mathbf{att}(Z) \cup (V(G) \setminus Z)$. It is straightforward to see that $(X_1, \ldots, X_{r+1})$ is a path decomposition of G of width at most $k-1$.

From now we assume that $\alpha(Z) < k$ and $|V(G) \setminus Z| > 1$. We show that the set Z can be extended by one vertex in such a way that the obtained set satisfies (i)–(iii). Let $u \in \mathbf{att}(Z) \cap V(C_0)$ and let v be an arbitrary neighbor of u in $V(G) \setminus Z$. We set $Z' = Z \cup \{v\}$ and let $X_{r+1} = \mathbf{att}(Z) \cup \{v\}$.

Because $V(G) \setminus Z' \neq \emptyset$ and G is connected, $\alpha(Z') \geq 1$. Clearly, $\alpha(Z') \leq \alpha(Z) + 1 \leq k$. Hence, (i) holds.

It is straightforward to verify that $(X_1, \ldots, X_{r+1})$ is a path decomposition of $G[Z']$ and $\mathbf{att}(Z') \subseteq \mathbf{att}(Z) \cup \{v\} \subseteq X_{r+1}$. The width of this decomposition is $\max\{w, t+1\}$ where w is the width of $(X_1, \ldots, X_r)$. Recall that $w \leq k-1$ and $t+1 = \alpha(Z) < k$. It means that (iii) is fulfilled.

It remains to show (ii). Let C_{t+1} be the subgraph of G with the unique vertex v. Clearly, $\mathbf{att}(Z') \subseteq V(C_0) \cup \ldots \cup V(C_{t+1})$ and G has an edge with one end-vertex in C_0 and another in C_i for all $i \in \{1, \ldots, t+1\}$. Since G is 2-connected, $G - u$ is connected, and G has a path that joins v with at least one of $C_1, \ldots, C_t$ that avoids C_0. Because $V(C_1) \cup \ldots \cup V(C_t)$ are in the same component of $G - V(C_0)$, we have that $V(C_1) \cup \ldots \cup V(C_{t+1})$ also are in the same component of $G - V(C_0)$. Notice that it can happen that not all C_i have vertices in $\mathbf{att}(Z')$. Let $\{C'_1, \ldots, C'_{t'}\} = \{C_i | V(C_i) \cap \mathbf{att}(Z') \neq \emptyset, 1 \leq i \leq t+1\}$. Because $V(C_1) \cup \ldots \cup V(C_{t+1})$ are in the same component of $G - V(C_0)$, $V(C'_1) \cup \ldots \cup V(C'_{t'})$ are in the same component of $G - V(C_0)$ too. Observe that since $|V(C_i) \cap \mathbf{att}(Z)| = 1$ for $i \in \{0, 1, \ldots, t\}$, we have $|V(C'_i) \cap \mathbf{att}(Z')| = 1$ for $i \in \{1, \ldots, t'\}$, $|V(C_0) \cap \mathbf{att}(Z')| \leq 1$, and $\mathbf{att}(Z') \subseteq V(C_0) \cup V(C'_1) \ldots \cup V(C'_{t'})$. We consider two cases.

Case 1. The vertex u has at least two neighbors in $V(G) \setminus Z$. Then C_0 has the unique vertex u in $\mathbf{att}(Z')$, and we have that $\alpha(Z') = t' + 1$ and (ii) holds for $C_0, C'_1, \ldots, C'_{t'}$.

Case 2. The vertex v is the unique neighbor of u in $V(G) \setminus Z$. Observe that since G is 2-connected, $t' \geq 2$ in this case. Consider the graph G' obtained from G by contracting edges of $C'_1, \ldots, C'_{t'}$ and denote by $x_1, \ldots, x_{t'}$ the vertices obtained from these graphs respectively. We have that $x_1, \ldots, x_{t'}$ are in the same component of $G' - V(C_0)$. We construct a spanning tree T for $\{x_1, \ldots, x_{t'}\}$ in $G' - V(C_0)$. Because $t' \geq 2$, T has at least two leaves. Without loss of generality we assume that x_1 is a leaf of T. Then $x_2, \ldots, x_{t'}$ and, consequently, $V(C'_2), \ldots, V(C'_{t'})$ are in the same component of $G' - (V(C_0) \cup \{x_1\})$ and $G - (V(C_0) \cup V(C'_1))$ respectively. We construct C'_0 by taking $C_0 \cup C'_1$ and adding an edge that joins C_0 and C'_1. Then $\mathbf{att}(Z') \subseteq V(C'_0) \cup V(C'_2) \ldots \cup V(C'_{t'})$ and G has an edge with one end-vertex in C'_0 and another in C'_i for all $i \in \{2, \ldots, t'\}$. Also $V(C'_2) \cup \ldots \cup V(C'_{t'})$ are in the same component of $G - V(C'_0)$. Because $V(C'_1) \cap \mathbf{att}(Z') \neq \emptyset$, $|V(C'_0) \cap \mathbf{att}(Z')| = 1$. Then $\alpha(Z') = t'$ and (ii) is fulfilled for $C'_0, C'_2, \ldots, C'_{t'}$. ◀

Observe that a non-empty proper subset Z of $V(G)$ that satisfies (i)–(iii) always exists, because for any vertex $z \in V(G)$, $Z = \{z\}$ satisfies (i)–(iii). Suppose that $\mathbf{pw}(G) \geq k$, and let $Z \subset V(G)$ be an inclusion-maximal non-empty proper subset of $V(G)$ that satisfies (i)–(iii). Then by Claim A, G has a minor H with the property that there is a vertex $w \in V(H)$ such that $d_H(w) \geq k$ and $H - w$ is a tree.

To complete the proof, it remains to observe that the proof of Claim A can be transformed to an algorithm that either constructs H, or produces a tree decomposition of G of width at most $k-1$, or increases Z by adding one vertex. In the last case the algorithm also modifies the subgraphs $C_0, \ldots, C_t$ and adds a new bag to the path decomposition. Initially we choose an arbitrary vertex z and set $Z = \{z\}$, $t = 0$ and C_0 has the unique vertex z. Since each iteration can be done in time $\mathcal{O}(m)$ and we have at most n iterations, we conclude that the algorithm runs in time $\mathcal{O}(nm)$. ◀

This combinatorial result is tight in the following sense. If $G = K_k$, then $\mathbf{pw}(G) = k-1$, and G has a minor H with the property that there is a vertex $w \in V(H)$ such that $d_H(w) \geq k$ and $H-w$ is a tree. But clearly G has no minors with a vertex of degree at least k. Theorem 4 gives us the following corollary.

▶ **Corollary 5.** *Let G be a 2-connected graph with n vertices and m edges. Let also k be a positive integer. If $\mathbf{pw}(G) \geq k$, then G has a set of terminals U of size at least k such that there are 2 completely independent spanning trees of U. Moreover, there is an algorithm that in time $\mathcal{O}(nm)$ either produces U and completely independent spanning trees T_1, T_2 of U, or constructs a path decomposition of G of width at most $k-1$.*

We conclude this section by the observation that the bounds obtained in Corollary 5 is almost tight. If $G = K_k$ with $k \geq 4$, we have $\mathbf{pw}(G) = k-1$, and there are two completely independent spanning trees of $V(G)$ where $|V(G)| = k+1$ and the number of terminals cannot be increased.

3.3 Proof of Theorem 1

In this section we give a proof of Theorem 1 by combining Lemma 3 and Corollary 5. However, we also need the following lemma which gives an algorithm for Independently 2-Connected k-Set on graphs of bounded treewidth.

▶ **Lemma 6.** *Let G be an n-vertex graph given together with its tree decomposition of width $\mathbf{tw}$. Then Independently 2-Connected k-Set on G can be solved in time $2^{\mathcal{O}(\mathbf{tw})}n^{\mathcal{O}(1)}$.*

A naive algorithm for Independently 2-Connected k-Set would run in time $\mathbf{tw}^{\mathcal{O}(\mathbf{tw})}n^{\mathcal{O}(1)}$. To obtain the desired running time, we use the idea of representative families introduced in [4] in our dynamic programming algorithm. By Lemma 2, we know that for Independently 2-Connected k-Set we need to find two edge disjoint trees (F_1, F_2) satisfying certain properties. Thus, if we take the intersection of the solution to some subgraph of the input graph we get two forests (F_1', F_2'). Let G be the input graph and H be an induced subgraph of G such that $|\partial(H)| \leq t$ where $\partial(H) = N(V(G) \setminus V(H))$. We call H, a t-boundaried graph. At every node of the tree decomposition one can associate a $t+1$ boundaried graph H of G. For H, we keep a family of partial solutions $\mathcal{P}$ that satisfies a following property. Given a solution (L_1, L_2) to Independently 2-Connected k-Set, there is a partial solution $(Q_1, Q_2) \in \mathcal{P}$ such that $(Q_1 \cup L_1^r, Q_2 \cup L_2^r)$ is also a solution. Here, $L_1^r = L_1 \setminus E(H)$ and $L_2^r = L_2 \setminus E(H)$. We use the ideas of matroids and representative families in order to bound the size of $\mathcal{P}$. One views each of the partial solution, (Q_1, Q_2), as a pair of forests in a graphic matroid of a clique on the vertex set $\partial(H)$. Thus these forests correspond to a pair of independent sets in graphic matroid. Furthermore, for every solution (L_1, L_2) to Independently 2-Connected k-Set, we view (L_1^r, L_2^r) as another pair of independent sets in graphic matroid of a clique on the vertex set $\partial(H)$. Now one observes that $(Q_1 \cup L_1^r, Q_2 \cup L_2^r)$ forms a pair of spanning tree of some induced subgraph of the clique. Once we have identified partial solutions as pairs of independent sets in a matroid one can show that the size of $\mathcal{P}$ is upper bounded by $2^{\mathcal{O}(t)}$. We finally give the proof of our main result.

Proof of Theorem 1. Let (G, k) be an input to Independently 2-Connected k-Set. Also assume that G has n vertices and m edges. We first compute all the blocks of G, say $B_1, \ldots, B_\ell$, in $\mathcal{O}(m+n)$ time. Now, by Lemma 3 we know that G is a yes-instance if and only if there exists an $i \in \{1, \ldots, \ell\}$ such that (B_i, k) is a yes-instance. Now on each B_i, we first apply Corollary 5 and in $\mathcal{O}(nm)$ time either produce a terminal set U and completely independent spanning trees T_1, T_2 of U, or construct a path decomposition of B_i of width at most $k-1$. In the former case we return U and completely independent spanning trees T_1, T_2 of U. In the later case we apply Lemma 6 and check whether (G, k) is a yes-instance to Independently 2-Connected k-Set. This completes the proof. ◀

4 Lower Bound on Kernelization

We proved that Independently 2-Connected k-Set is FPT. Hence, it is natural to ask whether this problem has a polynomial kernel. A parameterized problem Π is said to admit a kernel of size $f\colon \mathbb{N} \to \mathbb{N}$ if every instance (x, k) can be reduced in polynomial time to an equivalent instance with both size and parameter value bounded by $f(k)$. When $f(k) = k^{\mathcal{O}(1)}$ then we say that Π *admits a polynomial* kernel. The study of kernelization has recently been one of the main areas of research in parameterized complexity, yielding many important new contributions to the theory. The development of a framework for ruling out polynomial kernels under certain complexity-theoretic assumptions [1, 2, 5] has added a new dimension to the field and strengthened its connections to classical complexity.

Using the results by Bodlaender et al. [1], we show that it is unlikely even if we restrict ourself to 2-connected graph. We first give a few definitions required for our proof. A *composition algorithm* for a parameterized problem Π is an algorithm that receives as an input a sequence of instances $(I_1, k), \ldots, (I_t, k)$ of Π where each I_i is an input and k is a parameter, and in time polynomial in $\sum_{i=1}^{t} |I_i| + k$ produces an instance (I', k') of Π such that i) (I', k') is a YES-instance of Π if and only if (I_i, k) is a YES-instance for some $i \in \{1, \ldots, t\}$, and ii) k' is polynomial in k. If Π has a composition algorithm, then it is said that Π is *compositional*. Bodlaender et al. [1] proved the following theorem.

▶ **Theorem 7** ([1]). *If* Π *is a compositional parameterized problem such that the unparameterized version of* Π *is* NP*-complete, then* Π *has no polynomial kernel unless* $\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{poly}$.

It is easy to see that Independently 2-Connected k-Set is compositional for general (or connected) graphs. But by Lemma 3, it is sufficient to consider the problem for 2-connected graphs. Hence, we prove the following theorem.

▶ **Theorem 8.** Independently 2-Connected k-Set *has no polynomial kernel even for 2-connected graphs unless* $\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{poly}$.

Proof. As the unparameterized version of Independently 2-Connected k-Set is NP-complete for 2-connected graphs by the results of Hasunuma in [8], it is sufficient to show that Independently 2-Connected k-Set is compositional for 2-connected graphs.

Let $(G_1, k), \ldots, (G_t, k)$ be a sequence of instances of Independently 2-Connected k-Set where $G_1, \ldots, G_t$ are 2-connected, and we assume without loss of generality that $k \geq 3$. Let also $n_i = |V(G_i)| \geq 3$ for $i \in \{1, \ldots, t\}$, and denote by $v_1^i, \ldots, v_{n_i}^i$ the vertices of G_i for $i \in \{1, \ldots, t\}$. We construct G' as follows (see Fig. 1).

- For each $h \in \{1, \ldots, t\}$ and for each ordered pair (i, j) of distinct $i, j \in \{1, \ldots, n_h\}$, construct a copy $G_h^{(i,j)}$ of G_h; denote by $x_h^{(i,j)}$ and $y_h^{(i,j)}$ the vertices v_i^h and v_j^h of the copy $G_h^{(i,j)}$ of G_h respectively.

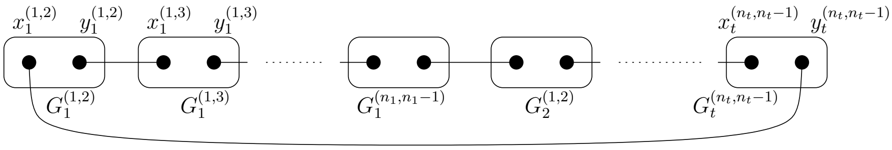

Figure 1 The construction of G'.

- For each $h \in \{1, \dots, t\}$, construct edges $y_h^{(i,j)} x_h^{(r,s)}$ for distinct ordered pairs $(i, j), (r, s)$ such that either $i = r$ and $s = j + 1$ or $r = i + 1$ and $j = n_h, s = 1$.
- For each $h \in \{1, \dots, t\}$, construct edges $y_h^{(n_h,n_h-1)} x_{h+1}^{(1,2)}$; we assume here that $x_{t+1}^{(1,2)} = x_1^{(1,2)}$.

We let $k' = 2k$. Notice that for all $x_h^{(i,j)}$ and $y_h^{(i,j)}$, G' has the unique edges that join these vertices with the vertices outside $G_h^{(i,j)}$. We call these edges by $x_h^{(i,j)}$ and $y_h^{(i,j)}$-*edges* respectively. Observe also that for all h, h' and $(i, j), (r, s)$, the graph G' has a $(y_h^{(i,j)}, x_{h'}^{(r,s)})$-path that contains $y_h^{(i,j)}$ and $x_{h'}^{(r,s)}$-edges.

It is straightforward to see that G' is 2-connected. We show that (G', k') is a YES-instance of Independently 2-Connected k'-Set if and only if (G_h, k) is a YES-instance for some $h \in \{1, \dots, t\}$.

Suppose that there is $h \in \{1, \dots, t\}$ such that G_h has a set of terminals U of size at least k such that there are two completely independent spanning trees F, T of U. Because $k \geq 3$, F and T have internal vertices. We choose such vertices denoted by v_i^h are v_j^h respectively. By Lemma 2, $i \neq j$. Denote by $F_h^{(i,j)}, T_h^{(i,j)}$ and $F_h^{(j,i)}, T_h^{(j,i)}$ the copies of F, T in $G_h^{(i,j)}$ and $G_h^{(j,i)}$ respectively. Let P be a $(y_h^{(i,j)}, x_h^{(j,i)})$-path in G' that contains $y_h^{(i,j)}$ and $x_h^{(j,i)}$-edges, and let Q be a $(y_h^{(j,i)}, x_h^{(i,j)})$-path in G' that contains $y_h^{(j,i)}$ and $x_h^{(i,j)}$-edges. Let T' be the tree obtained by taking the union of $T_h^{(i,j)}, T_h^{(j,i)}$ and P, and let F' be the tree obtained by taking the union of $F_h^{(i,j)}, F_h^{(j,i)}$ and Q. It remains to observe that F', T' are completely independent spanning trees of U' where U' is the union of the copies of U in $G_h^{(i,j)}$ and $G_h^{(j,i)}$. Since $|U'| = 2|U| \geq 2k$, we have that G a set of terminals U' of size at least k' such that there are two completely independent spanning trees F', T' of U'.

Suppose now that G a set of terminals U' of size at least k' such that there are two completely independent spanning trees F', T' of U'.

We claim that there are at most two $G_h^{(i,j)}$ that contain vertices of U'. To obtain a contradiction, assume that three distinct $G_{h_1}^{(i_1,j_1)}, G_{h_2}^{(i_2,j_2)}, G_{h_3}^{(i_3,j_3)}$ have vertices of U'. Then by the construction of G', there is $s \in \{1, 2, 3\}$ such that F' contains the $x_{h_s}^{(i_s,j_s)}$ and $y_{h_s}^{(i_s,j_s)}$-edges. Because F', T' are edge disjoint by Lemma 2, T' cannot contain any vertex of $G_{h_s}^{(i_s,j_s)}$; a contradiction. We consider two cases.

Case 1. The set U' contains vertices of the unique $G_h^{(i,j)}$. If F', T' do not include the $x_h^{(i,j)}$ and $y_h^{(i,j)}$-edges, then F', T' are subtrees of $G_h^{(i,j)}$. By taking the copies of F', T' in G_h, we have that G_h has a set of terminals of size at least $k' > k$ such that there are two completely independent spanning trees of the set. Suppose that one of the trees, say F', contains at least one of the $x_h^{(i,j)}$ and $y_h^{(i,j)}$-edges. Because F' is a minimal spanning tree of U', F' contains both the $x_h^{(i,j)}, y_h^{(i,j)}$-edges. Then F' has the unique $(y_h^{(i,j)}, x_h^{(i,j)})$-path P with these edges, and the internal vertices of P have degree two in F'. Then the forest obtained from F' by the deletion of the edges and the inner vertices of P has two components F_1 and F_2. Because $V(F') \cap U = (V(F_1) \cap U) \cup (V(F_2) \cap U)$ and $U_1 = (V(F_1) \cap U), U_2 = (V(F_2) \cap U)$

are disjoint, we can assume without loss of generality that $|U_1| \geq k$. Let F be the unique minimal spanning subtree of U_1 in F_1. Because F' contains the $x_h^{(i,j)}$ and $y_h^{(i,j)}$-edges, T' is a subgraph of $G_h^{(i,j)}$ by Lemma 2. Let T be be the unique minimal spanning subtree of U_1 in T'. We have that $G_h^{(i,j)}$ has the set of terminals U_1 of size at least k such that there are two completely independent spanning trees F, T of U_1. By taking the copies of F, T in G_h, we obtain that G_h has a set of terminals of size at least k such that there are two completely independent spanning trees of the set.

Case 2. The set U' contains vertices of two distinct $G_h^{(i,j)}, G_{h'}^{(r,s)}$. Let $U_1 = V(G_h^{(i,j)}) \cap U'$ and $U_2 = V(G_{h'}^{(r,s)}) \cap U'$. Because U_1, U_2 is a partition of U', we can assume without loss of generality that $|U_1| \geq k$. Notice that F', T' contain the $x_h^{(i,j)}, y_h^{(i,j)}, x_{h'}^{(r,s)}, y_{h'}^{(r,s)}$-edges, and the $x_h^{(i,j)}, y_{h'}^{(r,s)}$-edges (the $y_h^{(i,j)}, x_{h'}^{(r,s)}$-edges respectively) are in the same tree. We assume that F' contains the $x_h^{(i,j)}, y_{h'}^{(r,s)}$-edges and T' has the $y_h^{(i,j)}, x_{h'}^{(r,s)}$-edges. Then F' has the unique $(x_h^{(i,j)}, y_{h'}^{(r,s)})$-path Q and and T' has the unique $(y_h^{(i,j)}, x_{h'}^{(r,s)})$-path R, and the internal vertices of Q and R have degree two in F' and T' respectively. Then the forest obtained from F' by the deletion of the edges and the inner vertices of Q has exactly two components F_1, F_2, and it can be assumed that F_1 is a subgraph of $G_h^{(i,j)}$ and F_2 is a subgraph of $G_{h'}^{(r,s)}$. Notice that $U_1 \subseteq V(F_1)$, and let F be the unique spanning tree of U_1 in F_1. By the same arguments, the forest obtained from T' by the deletion of the edges and the inner vertices of R has exactly two components T_1, T_2, and it can be assumed that T_1 is a subgraph of $G_h^{(i,j)}$ and T_2 is a subgraph of $G_{h'}^{(r,s)}$. Again, $U_1 \subseteq V(F_1)$, and we consider the unique spanning tree T of U_1 in T_1. We have that $G_h^{(i,j)}$ has the set of terminals U_1 of size at least k such that there are two completely independent spanning trees F, T of U_1. By taking the copies of F, T in G_h, we obtain that G_h has a set of terminals of size at least k such that there are two completely independent spanning trees of the set.

In the both cases we have that there is $h \in \{1, \ldots, t\}$ such that (G_h, k) is a YES-instance of Independently 2-Connected k-Set, and it competes the proof. ◀

5 FPT algorithm for Independently s-Connected k-Set and a generalization

In this section we design an algorithm for Independently s-Connected k-Set. In fact, what we show is that this problem is is FPT when parameterized by $k + s$. We show that this problem can be reduced to checking existence of the bounded number of topological minors of bounded size. As the checking of existence of topological minors can be done in FPT-time by the recent results of Grohe et al. [6], we obtain the following theorem.

▶ **Theorem 9.** Independently s-Connected k-Set *is* FPT *when parameterized by* $s+k$.

Proof. If $k = 1$ or $s = 1$, then Independently s-Connected k-Set is trivial. If $k = 2$, then the problem can be solved in polynomial time by checking the existence of two vertices that can be joined by at least s internally vertex disjoint paths. Also if $s = 2$, then Independently s-Connected k-Set is FPT when parameterized by k by Theorem 1. Hence, we can assume that $s, k \geq 3$.

We prove the following two claims.

▶ **Claim B.** *If H is a topological minor of G such that (H, s, k) is a YES-instance of* Independently s-Connected k-Set*, then (G, s, k) is a YES-instance of* Independently s-Connected k-Set.

Proof of Claim B. Suppose that (H, s, k) is a YES-instance of Independently s-Connected k-Set for a topological minor H of G. Then there is a set of terminals $U \subseteq V(H)$ of size at least k and there are s completely independent spanning trees $T_1, \ldots, T_s$ of U in H. Since H is a topological minor of G, G has a subgraph H' such that H' can be obtained from H by a sequence of edge subdivisions. Let $T'_1, \ldots, T'_s$ be the trees obtained from $T_1, \ldots, T_s$ by applying these edge subdivisions to the edges of these trees. Denote by U' the set of vertices of G that correspond to the vertices of U in H'. It remains to observe that $T'_1, \ldots, T'_s$ are completely independent spanning trees of U' in G by Lemma 2, i. e., (G, s, k) is a YES-instance of Independently s-Connected k-Set. ◀

▶ **Claim C.** *If (G, s, k) is a YES-instance of* Independently s-Connected k-Set*, then G has a topological minor H with at most $sk + k - 2s$ vertices such that (H, s, k) is a YES-instance of* Independently s-Connected k-Set*.*

Proof of Claim C. Suppose that (G, s, k) is a YES-instance of Independently s-Connected k-Set. Then there is a set of terminals $U \subseteq V(G)$ of size exactly k and there are s completely independent spanning trees $T_1, \ldots, T_s$ of U in G. Let H be a subgraph of G that is the union of $T_1, \ldots, T_s$. Denote by H' the graph obtained from H by the recursive dissolutions of degree two vertices that have non-adjacent neighbors. Clearly, H' is a topological minor of G. Notice that because $s \geq 3$, the vertices of U are not dissolved, and we can dissolve only internal vertices of $T_1, \ldots, T_s$. Let $T'_1, \ldots, T'_s$ be the trees obtained from $T_1, \ldots, T_s$ respectively by these dissolutions. Then $T'_1, \ldots, T'_s$ are completely independent spanning trees of U in H' by Lemma 2, i. e., (H', s, k) is a YES-instance of Independently s-Connected k-Set.

To obtain the bound on the number of vertices of H', we show that for each T_i, all non-terminal internal vertices of degree two of T_i are dissolved. To obtain a contradiction, assume that at some step, we could not dissolve a vertex u of degree two. It can happen only if u has the neighbors x and y that are adjacent. Because T_i is a tree and the terminals are not dissolved, x and y are joined in some other tree T_j, i. e., $x, y \in V(T_i) \cap V(T_j)$. Moreover, x and y are joined in T_i, T_j by the unique (x, y)-paths P_i, P_j respectively such that the internal vertices of P_i, P_j have degree two in T_i, T_j respectively. By Lemma 2, $x, y \in U$. Because $k \geq 3$, each of x, y is an internal vertex of one of the trees $T_1, \ldots, T_s$ by Lemma 2. Since $s \geq 3$, either x or y is an internal vertex of at least two trees; a contradiction.

Thus, each T'_i has no non-terminal vertices of degree one or two. Therefore, because $|U| = k$, T'_i has at most $k-2$ internal vertices. Then the total number of internal vertices of $T'_1, \ldots, T'_s$ is at most $s(k-2)$, and the total number of vertices of H' is at most $s(k-2)+k$. ◀

Now we can solve Independently s-Connected k-Set as follows. We consider all $2^{\mathcal{O}(s^2k^2)}$ graphs H with at most $sk + k - 2s$ vertices. For each H, we solve Independently s-Connected k-Set using, e. g., brute force. If we obtain a yes-answer, then we check whether H is a topological minor of G by the algorithm of Grohe et al. [6]. If H is a topological minor of G, then (G, s, k) is a yes-instance of Independently s-Connected k-Set by Claim B. If we have a no-answer for all H, then Independently s-Connected k-Set for (G, s, k) has a no-answer by Claim C. ◀

A similar result can be obtained for the variant of the problem where a set of terminals is fixed. Formally, Independent Trees for a Set of Terminals ask for a graph G, positive integer s and a set U, whether there are s completely independent spanning trees of U in G. Using the same arguments as in the proof of Theorem 9, we can show the following.

▶ **Theorem 10.** Independent Trees for a Set of Terminals *is* FPT *when parameterized by* $s + |U|$.

6 Conclusions

In this paper we initiated parameterized complexity of a natural connectivity problem and designed several FPT algorithms for it. We conclude with several open questions.

- Is it possible to solve Independently s-Connected k-Set in time $2^{\mathcal{O}(k)}n^{\mathcal{O}(1)}$ for a fixed $s \geq 3$?
- What can be said about the approximability of Independently s-Connected k-Set? Is there a constant factor approximation algorithm for the problem for $s = 2$?
- We have shown that Independent Trees for a Set of Terminals is FPT when parameterized by $s + |U|$. Is it possible to obtain a more efficient algorithm for this problem? In particular, is it possible to solve the problem in single-exponential in $|U|$ for $s = 2$?

References

1. Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.
2. Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In Leonard J. Schulman, editor, *Proc. of the 42nd ACM Symp. on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5–8 June 2010*, pages 251–260. ACM, 2010.
3. Danny Dolev, Joseph Y. Halpern, Barbara Simons, and H. Raymond Strong. A new look at fault-tolerant network routing. *Inf. Comput.*, 72(3):180–196, 1987.
4. Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Efficient computation of representative sets with applications in parameterized and exact algorithms. In Chandra Chekuri, editor, *Proc. of the 25th Annual ACM-SIAM Symp. on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5–7, 2014*, pages 142–151. SIAM, 2014.
5. Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct pcps for NP. In Cynthia Dwork, editor, *Proc. of the 40th Annual ACM Symp. on Theory of Computing, Victoria, BC, Canada, May 17–20, 2008*, pages 133–142. ACM, 2008.
6. Martin Grohe, Ken-ichi Kawarabayashi, Dániel Marx, and Paul Wollan. Finding topological subgraphs is fixed-parameter tractable. In Lance Fortnow and Salil P. Vadhan, editors, *Proc. of the 43rd ACM Symp. on Theory of Computing, STOC 2011, San Jose, CA, USA, 6–8 June 2011*, pages 479–488. ACM, 2011.
7. Toru Hasunuma. Completely independent spanning trees in the underlying graph of a line digraph. *Discrete Mathematics*, 234(1-3):149–157, 2001.
8. Toru Hasunuma. Completely independent spanning trees in maximal planar graphs. In Ludek Kucera, editor, *Graph-Theoretic Concepts in Computer Science, 28th International Workshop, WG 2002, Cesky Krumlov, Czech Republic, June 13–15, 2002, Revised Papers*, volume 2573 of *Lecture Notes in Computer Science*, pages 235–245. Springer, 2002.
9. Toru Hasunuma and Chie Morisaka. Completely independent spanning trees in torus networks. *Networks*, 60(1):59–69, 2012.
10. Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.
11. Alon Itai and Michael Rodeh. The multi-tree approach to reliability in distributed networks. *Inf. Comput.*, 79(1):43–59, 1988.
12. Ferenc Péterfalvi. Two counterexamples on completely independent spanning trees. *Discrete Math.*, 312(4):808–810, 2012.