

Parameterized Complexity of Secluded Connectivity Problems*

Fedor V. Fomin^{†‡} Petr A. Golovach^{†‡} Nikolay Karpov[‡]
Alexander S. Kulikov[‡]

October 14, 2016

Abstract

The SECLUDED PATH problem models a situation where sensitive information has to be transmitted between a pair of nodes along a path in a network. The measure of the quality of a selected path is its *exposure cost*, which is the total cost of vertices in its closed neighborhood. The task is to select a *secluded* path, i.e., a path with a small exposure cost. Similarly, the SECLUDED STEINER TREE problem is to find a tree in a graph connecting a given set of terminals such that the exposure cost of the tree is minimized. In this paper we present a systematic study of the parameterized complexity of SECLUDED STEINER TREE. In particular, we establish the tractability of SECLUDED PATH being parameterized by “above guarantee” value, which in this case is the length of a shortest path between vertices. We also show how to extend this result for SECLUDED STEINER TREE, in this case we parameterize above the size of an optimal Steiner tree and the number of terminals. We also consider various parameterization of the problems such as by the treewidth, the size of a vertex cover, feedback vertex set, or the maximum vertex degree and establish kernelization complexity of the problem subject to different choices of parameters.

1 Introduction

SECLUDED PATH and SECLUDED STEINER TREE problems were introduced in Chechik et al. in [10]. In the SECLUDED PATH problem, for given

*A preliminary version of this paper appeared as an extended abstract in the proceedings of FSTTCS 2015 [15]. The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 267959 and the Government of the Russian Federation (grant 14.Z50.31.0030).

[†]Department of Informatics, University of Bergen, Norway.

[‡]Steklov Institute of Mathematics at St.Petersburg, Russian Academy of Sciences, Russia.

vertices s and t of a graph G , the task is to find an (s, t) -path with the minimum *exposure*, i.e. a path P such that the number of vertices from P plus the number of vertices of G adjacent to vertices of P is minimized. The name secluded comes from the setting where one wants to transfer confident information over a path in a network which can be intercepted either while passing through a vertex of the path or from some adjacent vertex. Thus the problem is to select a secluded path minimizing the risk of interception of the information. When instead of connecting two vertices one needs to connect a set of terminals, we arrive naturally at the SECLUDED STEINER TREE.

More precisely, SECLUDED STEINER TREE is the following problem.

SECLUDED STEINER TREE

Input: A graph G with a cost function $\omega: V(G) \rightarrow \mathbb{N}$ with $\max_{v \in V(G)} \omega(v) \leq W$, a set $S = \{s_1, \dots, s_p\} \subseteq V(G)$ of terminals, and non-negative integers k and C .

Question: Is there a connected subgraph T of G with $S \subseteq V(T)$ such that $|N_G[V(T)]| \leq k$ and $\omega(N_G[V(T)]) \leq C$?

If $\omega(v) = 1$ for each $v \in V(G)$ and $C = k$, then we have an instance of SECLUDED STEINER TREE without costs; respectively, we omit ω and C whenever we consider such instances. We say that T with the required properties is a *solution*, and we call $|N_G[V(T)]|$ the *exposure size*, and $\omega(N_G[V(T)])$ is the *exposure cost*. We also refer to a solution T as a *secluded Steiner tree*.

Clearly, it can be assumed that T is a tree, and thus the problem can be seen as a variant of the classical STEINER TREE problem. For the special case $p = 2$, we call the problem SECLUDED PATH. Respectively, we refer to a solution as a *secluded path*.

Previous work. The study of the secluded connectivity was initiated by Chechik et al. [9, 10] who showed that the decision version of SECLUDED PATH without costs is NP-complete. Moreover, for the optimization version of the problem, it is hard to approximate within a factor of $\mathcal{O}(2^{\log^{1-\varepsilon} n})$, where n is the number of vertices in an input graph, for any $\varepsilon > 0$ (under an appropriate complexity assumption) [10]. Chechik et al. [10] also provided several approximation and parameterized algorithms for SECLUDED PATH and SECLUDED STEINER TREE. Interestingly, when there are no costs, SECLUDED PATH is solvable in time $\Delta^\Delta \cdot n^{\mathcal{O}(1)}$, where Δ is the maximum vertex degree and thus is FPT being parameterized by Δ . Chechik et al. [10] also showed that SECLUDED STEINER TREE is FPT when parameterized by the treewidth of the input graph.

Johnson et al. [22] obtained several approximation results for SECLUDED PATH and showed that the problem with costs is NP-hard for subcubic graphs improving a previous result of Chechik et al. [10] for graphs of maximum

degree 4.

The problems related to secluded path and connectivity under different names were considered by several authors. Motivated by secure communications in wireless ad hoc networks, Gao et al. [18] introduced the very similar notion of the thinnest path. The motivation of Gilbers [20], who introduced the problem under the name of the minimum witness path, came from the study of art gallery problems.

Our results. In this paper we initiate the systematic study of both problems from the Parameterized Complexity perspective and obtain the following results.

In Section 3, we start with a proof that when parameterized by the exposure size k SECLUDED PATH and SECLUDED STEINER TREE are FPT. In particular, for a graph G with n vertices and m edges, we give algorithms solving

- SECLUDED PATH in time $\mathcal{O}(3^{k/3} \cdot (n + m) \log W)$, and
- SECLUDED STEINER TREE $\mathcal{O}(2^k k^2 \cdot (n + m) \log W)$.

We complement the algorithmic results by showing that a “stronger” parameterization by the solutions size is hard:

- SECLUDED PATH without costs is W[1]-hard when parameterized by the length of a solution path.

In Section 4, we consider the “above guarantee” parameterizations of both problems. Let s_1, \dots, s_p be terminal vertices of a graph G . Then a connected subgraph T of G of minimum size such that $s_1, \dots, s_p \in V(T)$ is called a *Steiner tree* for the terminals s_1, \dots, s_p . If $p = 2$, then a Steiner tree is a shortest (s_1, s_2) -path. Let ℓ be the size (the number of vertices) of a Steiner tree. Then the size of every secluded Steiner tree is at least ℓ and it is natural to ask about the complexity of the problem parameterized by the difference $k - \ell$. However, as we show in Theorem 8,

- SECLUDED STEINER TREE is co-W[1]-hard when parameterized by $k - \ell$.

Let us recall that while the STEINER TREE problem is well known to be NP-complete [23], it is FPT parameterized by the number of terminals. In 1971 Dreyfus and Wagner [13] proved that the problem can be solved in time $3^p \cdot n^{\mathcal{O}(1)}$. The best known FPT-algorithms for STEINER TREE run in time $2^p \cdot n^{\mathcal{O}(1)}$ and are due to Björklund et al. [4] and Nederlof [25]. (The first algorithm demands exponential in p space and the latter uses polynomial space.) In Section 4 we show that SECLUDED PATH and SECLUDED STEINER TREE are FPT when the problems are parameterized by $k - \ell + p$. In particular, we obtain the following results

- SECLUDED PATH is solvable in time $\mathcal{O}(2^{k-\ell} \cdot (n+m) \log W)$, and
- SECLUDED STEINER TREE can be solved in time $2^{\mathcal{O}(k-\ell+p)} \cdot nm \cdot \log W$ by a true-biased Monte-Carlo algorithm. This algorithm can be derandomized in time $2^{\mathcal{O}(k-\ell+p)} \cdot nm \log n \cdot \log W$.

In Section 5, we provide a thorough study of the kernelization of the problem from the structural parameterization perspective. We consider parameterizations by the exposure size k , the treewidth $\mathbf{tw}(G)$, maximum degree $\Delta(G)$ and the sizes of a vertex cover $\mathbf{vc}(G)$ and a feedback vertex set $\mathbf{fvs}(G)$ of an input graph G .

We show in Theorem 9 that

- SECLUDED PATH without costs admits no polynomial kernel unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$ when parameterized by $k + \mathbf{tw}(G) + \Delta(G)$.

This complements the FPT algorithmic findings of Chechik et al. [10] for graphs of bounded treewidth and of bounded maximum vertex degree. Notice that Theorem 9 implies that SECLUDED PATH without costs has no polynomial kernel unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$ when parameterized only by k or $\mathbf{tw}(G)$. We strengthen the latter by showing that

- SECLUDED PATH without costs has no polynomial kernel unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$ when parameterized by $\mathbf{vc}(G)$.

On the other hand, a “weaker” parameterization by the size of a feedback vertex set and the exposure size brings us to a polynomial kernel.

- SECLUDED STEINER TREE admits a kernel with $\mathcal{O}(k^3 \cdot \mathbf{fvs}(G))$ vertices.

2 Basic definitions and preliminaries

We consider only 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 $E(G)$. Throughout the paper we typically use n and m to denote the number of vertices and edges respectively.

For a set of vertices $U \subseteq V(G)$, $G[U]$ denotes the subgraph of G induced by U . For a vertex v , we denote by $N_G(v)$ its (*open*) *neighborhood*, that is, the set of vertices which are adjacent to v , and for a set $U \subseteq V(G)$, $N_G(U) = (\cup_{v \in U} N_G(v)) \setminus U$. The *closed neighborhood* $N_G[v] = N_G(v) \cup \{v\}$. Respectively, $N_G[U] = N_G(U) \cup U$. For a set $U \subseteq V(G)$, $G - U$ denotes the subgraph of G induced by $V(G) \setminus U$. If $U = \{u\}$, we write $G - u$ instead of $G - \{u\}$. The *degree* of a vertex v is denoted by $d_G(v) = |N_G(v)|$. We say that a vertex v is *pendant* if $d_G(v) = 1$. A vertex v of a connected graph G with at least 2 vertices is a *cut vertex* if $G - v$ is disconnected. A connected

graph G is *biconnected* if it has at least 2 vertices and has no cut vertices. A *block* of a connected graph G is an inclusion-maximal biconnected subgraph of G . A block is *trivial* if it has exactly 2 vertices. We say that vertex set X is connected if $G[X]$ is connected.

A *tree decomposition* of a graph G is a pair (\mathcal{B}, T) where T is a tree and $\mathcal{B} = \{B_i \mid i \in V(T)\}$ is a collection of subsets (called *bags*) of $V(G)$ such that

- i) $\bigcup_{i \in V(T)} B_i = V(G)$,
- ii) for each edge $xy \in E(G)$, $x, y \in B_i$ for some $i \in V(T)$, and
- iii) for each $x \in V(G)$ the set $\{i \mid x \in B_i\}$ induces a connected subtree of T .

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

A set $U \subseteq V(G)$ is a *vertex cover* of G if for any edge uv of G , $u \in U$ or $v \in U$. The *vertex cover number* $\mathbf{vc}(G)$ is the size of a minimum vertex cover.

A set $U \subseteq V(G)$ is a *feedback vertex set* of G if $G - U$ is acyclic, that is, $G - U$ is a forest. We denote by $\mathbf{fvs}(G)$ the minimum size of a feedback vertex set of G .

Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. One dimension is the input size n and another one is a parameter k . Formally speaking, a parameterized problem is a set $P \subseteq \Sigma^* \times \mathbb{N}$, where Σ^* is a set of strings over a finite alphabet Σ . Respectively, for $x \in \Sigma^*$ and $k \in \mathbb{N}$, (x, k) is a *yes-instance* of the problem if $(x, k) \in P$. We refer to the recent books of Cygan et al. [11] and Downey and Fellows [12] for detailed introductions to parameterized complexity.

It is said that a problem is *fixed parameter tractable* (or FPT), if it can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some function f . A *kernelization* for a parameterized problem is a polynomial algorithm that maps each instance (x, k) with the input x and the parameter k to an instance (x', k') such that i) (x, k) is a yes-instance if and only if (x', k') is a yes-instance of the problem, and ii) $|x'| + k'$ is bounded by $f(k)$ for a computable function f . The output (x', k') is called a *kernel*. The function f is said to be a *size* of a kernel. Respectively, a kernel is *polynomial* if f is polynomial.

While a parameterized problem is FPT if and only if it has a kernel, it is widely believed that not all FPT problems have polynomial kernels. In particular, Bodlaender et al. [5, 6] introduced techniques that allow to show that a parameterized problem has no polynomial kernel unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$.

The cross-composition technique was introduced by Bodlaender, Jansen and Kratsch [6]. We need the following additional definitions (see [6]).

Let Σ be a finite alphabet. An equivalence relation \mathcal{R} on the set of strings Σ^* is called a *polynomial equivalence relation* if the following two conditions hold:

- i) there is an algorithm that given two strings $x, y \in \Sigma^*$ decides whether x and y belong to the same equivalence class in time polynomial in $|x| + |y|$,
- ii) for any finite set $S \subseteq \Sigma^*$, the equivalence relation \mathcal{R} partitions the elements of S into a number of classes that is polynomially bounded in the size of the largest element of S .

Let $L \subseteq \Sigma^*$ be a language, let \mathcal{R} be a polynomial equivalence relation on Σ^* , and let $\mathcal{Q} \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An *OR-cross-composition of L into \mathcal{Q}* (with respect to \mathcal{R}) is an algorithm that, given t instances $x_1, x_2, \dots, x_t \in \Sigma^*$ of L belonging to the same equivalence class of \mathcal{R} , takes time polynomial in $\sum_{i=1}^t |x_i|$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that:

- i) the parameter value k is polynomially bounded in $\max\{|x_1|, \dots, |x_t|\} + \log t$,
- ii) the instance (y, k) is a yes-instance for \mathcal{Q} if and only if at least one instance x_i is a yes-instance for L for $i \in \{1, \dots, t\}$.

It is said that L *OR-cross-composes into* \mathcal{Q} if an OR-cross-composition algorithm exists for a suitable relation \mathcal{R} .

In particular, Bodlaender, Jansen and Kratsch [6] proved the following theorem.

Theorem 1 ([6]). *If an NP-hard language L OR-cross-composes into the parameterized problem \mathcal{Q} , then \mathcal{Q} does not admit a polynomial kernelization unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$.*

We use randomized algorithms for our problems. Recall that a *Monte Carlo algorithm* is a randomized algorithm whose running time is deterministic, but whose output may be incorrect with a certain (typically small) probability. A Monte-Carlo algorithm is *true-biased* (*false-biased* respectively) if it always returns a correct answer when it returns a yes-answer (a no-answer respectively).

3 FPT-algorithms for the problems parameterized by the exposure size

In this section we consider SECLUDED PATH and SECLUDED STEINER TREE problems parameterized by the exposure size of a solution, i.e., by k . We

also show how these parameterized algorithms can be used to design faster exact exponential algorithms. The following proposition from [17] will be useful for us.

Proposition 1 ([17]). *Let G be a graph. For every $v \in V(G)$, and $b, f \geq 0$, the number of connected vertex subsets $B \subseteq V(G)$ such that*

- (i) $v \in B$,
- (ii) $|B| = b + 1$, and
- (iii) $|N_G(B)| = f$,

is at most $\binom{b+f}{b}$. Moreover, all such subsets can be enumerated in time $\mathcal{O}(\binom{b+f}{b} \cdot (n+m) \cdot b \cdot (b+f))$.

We start with SECLUDED PATH.

Theorem 2. SECLUDED PATH is solvable in time $\mathcal{O}(3^{k/3} \cdot n \log W)$, where W is the maximum value of ω on an input graph G .

Proof. Let us observe first that if there is a secluded path, then there is a secluded path that is an induced path—shortcutting a path cannot increase the size of its neighbourhood. We give an algorithm that enumerates all induced paths P from u to v such that $|N_G[V(P)]| \leq k$ in time $\mathcal{O}(3^{k/3} \cdot n)$ for a graph G with n vertices. Then picking a secluded path of minimum cost will complete the proof.

The algorithm is based on the standard branching ideas. If $|N_G[u]| > k$ the algorithm reports that no such path exist and stops. If $|N_G[u]| \leq k$ and $u = v$ the algorithm outputs the path consisting of the single vertex u . Otherwise a path from u to v must go through one of the neighbors of u . Since we are looking for an induced path it must never return to a vertex from $N_G[u]$. This allows us to branch as follows. For each $w \in N_G(u)$, we check recursively whether the graph $G_w = (G \setminus N_G[u]) \cup \{w\}$ contains an induced path Q from w to v such that $|N_{G_w}[Q]| \leq k - |N_G(u)|$. This way we get the following recurrence on the number of nodes $t(k)$ in the corresponding recursion tree. If $u = v$, then there is only one path from u to v , and $t(k) \leq 1$. If $u \neq v$, then $t(k) \leq d \cdot t(k-d)$, where $d = |N_G(u)|$. This is a well known recurrence implying that $t(k) = \mathcal{O}(3^{k/3})$ (see, e.g., the analysis of the algorithm enumerating all maximal independent sets in Chapter 1 of [16]). Note that we spend only a linear time $\mathcal{O}(n)$ in each vertex of the recursion tree. Since the length of each path P can be computed in time $\mathcal{O}(n \log W)$, we can find a path of minimum cost in time $\mathcal{O}(3^{k/3} n \log W)$. Therefore, the total running time is $\mathcal{O}(3^{k/3} \cdot n \log W)$. \square

For SECLUDED STEINER TREE we prove the following theorem.

Theorem 3. SECLUDED STEINER TREE can be solved in time $\mathcal{O}(2^k k^2 \cdot (n + m) \log W)$, where W is the maximum value of ω on an input graph G .

Proof. By Proposition 1, the number of connected sets T of size b containing s_1 and such that $|N_G[T]| = b + f$, does not exceed $\binom{b+f}{b}$. Since $b + f \leq k$, we have that the number of such sets does not exceed

$$\sum_{b+f \leq k} \binom{b+f}{b} = \sum_{r=0}^k \sum_{b=0}^r \binom{r}{b} = \sum_{r=0}^k 2^r = 2^{k+1} - 1.$$

By Proposition 1, all such sets $N_G[T]$ can be enumerated in time

$$\mathcal{O}\left(\sum_{b+f \leq k} \binom{b+f}{b} \cdot b(b+f) \cdot (n+m)\right).$$

Since $b, f \leq k$,

$$\sum_{b+f \leq k} \binom{b+f}{b} \cdot b(b+f) \leq k^2 \sum_{b+f \leq k} \binom{b+f}{b} \leq k^2 \cdot 2^{k+1},$$

these sets can be enumerated in time $\mathcal{O}(2^k k^2 \cdot (n+m))$. While enumerating sets $N_G[T]$, we disregard sets not containing all terminal vertices. Finally, in time $\mathcal{O}(2^k \log W)$ we select the set of minimum cost. \square

Parameterized algorithms for SECLUDED PATH and SECLUDED STEINER TREE combined with a brute-force procedure imply the following exact exponential algorithms for the problems.

Theorem 4. On an n -vertex graph, SECLUDED PATH is solvable in time $\mathcal{O}(1.3896^n \cdot \log W)$ and SECLUDED STEINER TREE is solvable in time $\mathcal{O}(1.7088^n \cdot \log W)$, where W is the maximum value of ω on an input graph G .

Proof. By Theorem 2, SECLUDED PATH is solvable in time $3^{k/3} \cdot n \log W$. On the other hand, we also can solve the problem by the brute-force procedure checking every set X of size $n - i$. Notice that $V(G) \setminus X$ contains the closed neighborhood of a secluded path if and only if $V(G) \setminus N_G[X]$ is connected, contains both terminal vertices, $|V(G) \setminus X| \leq k$ and $\omega(V(G) \setminus X) \leq C$. Clearly, and these conditions can be checked in polynomial time. The brute-force procedure takes time $\binom{n}{n-i} \cdot n^{\mathcal{O}(1)} \log W$.

It is well-known (see, e.g., [16]) that if $1/2 < \varepsilon < 1$, then for $n/2 \leq \varepsilon n \leq i \leq n$, it holds that $\binom{n}{i} \leq 2^{H(\varepsilon)n}$, where $H(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the *entropy function*. In particular, if $\varepsilon \geq 0.8983$ and $\varepsilon n \leq i \leq n$,

$$\binom{n}{n-i} = \binom{n}{i} \leq 2^{H(\varepsilon)n} < 3^{\varepsilon n/3}.$$

Thus for all integers i between $0.8983 \cdot n$ and n , we enumerate sets of size $n - i$, while for all integers i between 1 and $0.8983 \cdot n$ we use Theorem 2 to find if there is a solution of the exposure size at most i . The running time of the algorithm is dominated by $\mathcal{O}(3^{\frac{0.8983n}{3}} \cdot \log W) = \mathcal{O}(1.3896^n \cdot \log W)$.

Similarly, we use parameterized time $2^k \cdot n^{\mathcal{O}(1)} \log W$ algorithm from Theorem 3 for SECLUDED STEINER TREE and balance it with the brute-force procedure checking for every set X of size $n - i$, whether $V(G) \setminus X$ is the closed neighbourhood of a secluded Steiner tree T . For each such set X , we check in polynomial time whether $V(G) \setminus N_G[X]$ is connected, contains all terminal vertices, $|V(G) \setminus X| \leq k$ and $\omega(V(G) \setminus X) \leq C$. The brute-force runs in time $\binom{n}{n-i} \cdot n^{\mathcal{O}(1)} \log W$.

For $\varepsilon \geq 0.77291$, we have that

$$\binom{n}{n-i} = \binom{n}{i} \leq 2^{H(\varepsilon)n} < 2^{\varepsilon n},$$

if $\varepsilon n \leq i \leq n$. Thus for all integers i between $0.77291 \cdot n$ and n , we enumerate sets of size $n - i$, while for all integers i between 1 and $0.77291 \cdot n$ we use Theorem 3 to find if there is a solution of the exposure size at most i . The running time of this algorithm is $\mathcal{O}(2^{0.77291n} \cdot \log W) = \mathcal{O}(1.7088^n \cdot \log W)$. \square

Recall that we defined the exposure size of a solution T for an instance of SECLUDED STEINER TREE as $|N_G(V(T))|$ and we consider SECLUDED STEINER TREE parameterized by the solution size in this section. It is natural to ask what can be said if we parameterize the problem by $|V(T)|$, that is, by the solutions size instead of $|N_G(V(T))|$. In the conclusion of this section we show that the problem is hard for this parameterization.

Theorem 5. SECLUDED PATH *without costs is* W[1]-hard when parameterized by the length of a solution path.

Proof. We reduce the variant of the MULTICOLORED CLIQUE problem, where an input graph is required to be regular.

REGULAR MULTICOLORED CLIQUE	Parameter: k
Input: A regular graph G , a positive integer k , and a partition V_1, \dots, V_k of $V(G)$.	
Question: Does G has a clique K such that $ V_i \cap K = 1$ for $i \in \{1, \dots, k\}$?	

It was proved by Cai in [7] that the CLIQUE problem that asks whether a graph G has a clique of size k remains W[1]-hard on regular graphs when parameterized by k . Combining this results with the standard parameterized reduction from CLIQUE to MULTICOLORED CLIQUE (see [14, 26]) we immediately obtain that REGULAR MULTICOLORED CLIQUE is W[1]-hard.

Let (G, k, V_1, \dots, V_k) be an instance of REGULAR MULTICOLORED CLIQUE, and assume that G is a $d \geq 1$ -regular n -vertex graph and each V_i is an independent set. We construct the graph H as follows.

- i) Construct copies of V_1, \dots, V_k .
- ii) Construct vertices u_0, \dots, u_k and for each $i \in \{1, \dots, k\}$, make u_{i-1} and u_i adjacent to all the vertices of V_i .
- iii) For each edge xy of G , construct a vertex w_{xy} and make it adjacent to the copies of x and y in $V_1 \cup \dots \cup V_k$.
- iv) Construct $p = n + k + dk$ vertices y_1, \dots, y_p and make each of them adjacent to w_{xy} for $xy \in E(G)$.

We set $\ell = 2k$ and $r = n + k + 1 + dk - k(k-1)/2$. Let $s = u_0$ and $t = u_k$.

We claim that there is an (s, t) -path P in H of length at most ℓ such that $|N_H[V(P)]| \leq r$ if and only if G has a clique K such that $|V_i \cap K| = 1$ for $i \in \{1, \dots, k\}$.

Let $K = \{v_1, \dots, v_k\}$ be a clique of G such that $v_i \in V_i$. Consider the copies of v_1, \dots, v_k in H and let $P = u_0 v_1 u_1 \dots u_{k-1} v_k u_k$. It is straightforward to verify that $|N_H[V(P)]| = r$.

Suppose that H has a (s, t) -path P of length at most ℓ such that $|N_H[V(P)]| \leq r$. Notice that $w_{xy} \notin V(P)$ for any $xy \in E(G)$, because $d_H(w_{xy}) > r$. Hence, $P = u_0 v_1 u_1 \dots u_{k-1} v_k u_k$ for some $v_i \in V_i$ for $i \in \{1, \dots, k\}$. Let F be a subgraph of G induced by the copies of v_1, \dots, v_k . Since H is d -regular, $|N_H[V(P)]| = k + 1 + n + dk - |E(F)| \leq r = n + k + 1 + dk - k(k-1)/2$. Therefore, $|E(F)| \geq k(k-1)/2$. It means that F is a complete graph, that is, the copies of v_1, \dots, v_k compose a clique of size k in G . \square

4 FPT-algorithms for the problems parameterized above the guaranteed value

In this section we show that SECLUDED PATH and SECLUDED STEINER TREE are FPT when the problems are parameterized by $r + p$ where $r = k - \ell$ and ℓ is the size of a Steiner tree for S .

Theorem 6. SECLUDED PATH is solvable in time $\mathcal{O}(2^{k-\ell} \cdot (n + m) \log W)$, where ℓ is the length of a shortest (u, v) -path for $\{u, v\} = S$ and W is the maximum value of ω on an input graph G .

Proof. The proof of this theorem is very similar to the proof of Theorem 2. For an integer h , we enumerate in the graph G all induced paths P from u to v of length at most h such that $|N_G[V(P)]| \leq k - h$. The only difference with Theorem 2 is that this time we bound the running time of the algorithm as a function of $k - h$.

If $|N_G(u)| > k - h$ the algorithm reports that no such path exist and stops. If $|N_G[u]| \leq k - h$ and $u = v$ the algorithm outputs the path consisting

of the single vertex u . Otherwise, we branch by checking recursively for each $w \in N_G(u)$, whether the graph $G_w = (G \setminus N_G[u]) \cup \{w\}$ contains an induced path Q from w to v of length at most $h - 1$ such that $|N_{G_w}(Q)| \leq k - |N_G(u)| - (h - 1)$. This way we get the following recurrence on the number of nodes $T(k - h)$ in the corresponding recursion tree. If $u = v$, then there is only one path from u to v , and $T(k - h) \leq 1$. If $u \neq v$, then

$$T(k - h) \leq d \cdot T(k - d - h + 1),$$

where $d = |N_G(u)|$. It is easy to show, that $T(k - h) = \mathcal{O}(2^{k-h})$. \square

We need some structural properties of solutions of SECLUDED STEINER TREE. We start with an auxiliary lemma bounding the number of vertices of degree at least three in the subgraph of G induced by a solution as well as the number of their neighbors in this subgraph.

Lemma 1. *Let G be a connected graph and $S \subseteq V(G)$, $p = |S|$. Let F be an inclusion minimal connected induced subgraph of G such that $S \subseteq V(F)$ and $X = \{v \in V(F) \mid d_F(v) \geq 3\} \cup S$ (see Fig. 1). Then (i) $|X| \leq 4p - 6$, and (ii) $|N_F(X)| \leq 4p - 6$.*

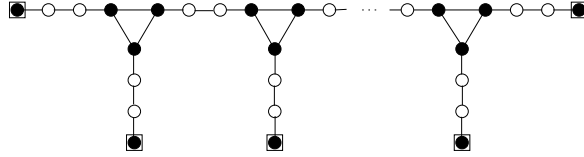


Figure 1: An example of F ; the terminals are show as squares, the vertices of X are black and the remaining vertices are white.

Proof. Let \mathcal{B} be the set of blocks of F . Consider a bipartite graph T with the bipartition $(V(F), \mathcal{B})$ of the vertex set such that $v \in V(F)$ and $b \in \mathcal{B}$ are adjacent if and only if v is a vertex of b . Notice that T is a tree. Recall that the *vertex dissolution* operation for a vertex v of degree 2 deletes v together with incident edges and replaces them by the edge joining the neighbors of v . Denote by T' the tree obtained from T by consequent dissolving all vertices of T of degree 2 that are not in S . Denote by L the set of leaves of T . By the minimality of F , $L \subseteq S$. Let $q_1 = |L| \leq p$, and let q_2 be the number of degree 2 vertices and q_3 be the number of vertices of degree at least 3 in T . Clearly, $q_1 + 2q_2 + 3q_3 \leq 2|E(T)| = 2(q_1 + q_2 + q_3 - 1)$. Then $q_3 \leq q_1 - 2 \leq p - 2$. We have that $|\{v \in V(T) \mid d_T(v) \geq 3\} \cup S| \leq q_3 + p \leq 2p - 2$ and $|V(T')| \leq 2p - 2$. Observe that if $d_F(v) \geq 3$ for $v \in V(F) \setminus S$, then v is a cut vertex of F and either v is included in at least 3 blocks of F , or v is in a block of size at least 3. In the second case, v is adjacent to a vertex $b \in \mathcal{B}$ of T with degree at least 3. It implies that $|X| \leq 2|E(T')| = 2(|V(T')| - 1) \leq 4p - 6$ and we have (i). To show (ii), observe that $|N_F(X)| \leq 2|E(T')| \leq 4p - 6$. \square

The example on Fig. 1 shows that the bounds (i) and (ii) in Lemma 1 are tight. The following lemma provides a bound on the number of vertices of a tree that have neighbors outside the tree.

Lemma 2. *Let G be a connected graph and $S \subseteq V(G)$, $p = |S|$. Let ℓ be the size of a Steiner tree for S and r be a positive integer. Suppose that T (see Fig. 2) is an inclusion minimal subgraph of G such that T is a tree spanning S and $|N_G[V(T)]| \leq \ell + r$. Then for $Y = N_G(V(T))$, $|N_G(Y) \cap V(T)| \leq 4p + 2r - 5$.*

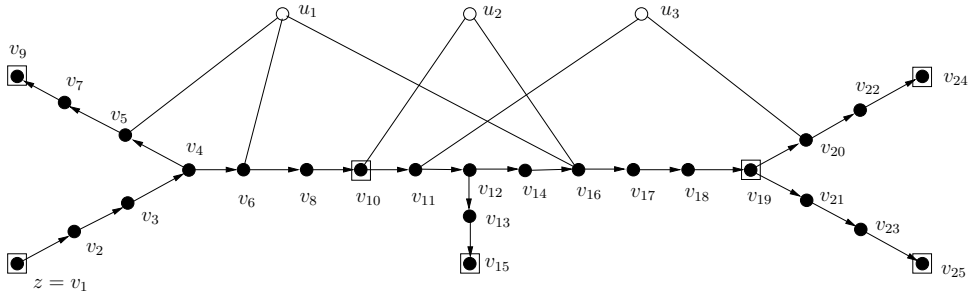


Figure 2: An example of T (the vertices are black) rooted in z with its neighborhood $Y = N_G(V(T)) = \{u_1, u_2, u_3\}$ (the vertices are white); the vertices of T are numbered v_1, \dots, v_{25} according to their ordering \prec with respect to the root $z = v_1$, the vertices of the set of terminals $S = \{v_1, v_9, v_{10}, v_{15}, v_{19}, v_{24}, v_{25}\}$ are shown by squares, $N_G(Y) \cap V(T) = \{v_5, v_6, v_{10}, v_{11}, v_{16}, v_{20}\}$ and the set of vertices of degree at least 3 in T $D = \{v_4, v_{12}, v_{19}\}$.

Proof. Denote by L the set of leaves of T and by D the set of vertices of degree at least 3 in T . Clearly, $L \subseteq S$. We select a leaf z of T as the *root* of T . The selection of a root defines a parent-child relation on T . We order the vertices of T by the increase of their distances to z in T ; the vertices on the same distance are ordered arbitrarily. Denote the obtained linear order by \prec . See Fig. 2 for an example. For each $u \in Y$, denote by $x(u)$ the unique minimum vertex in $N_G(u) \cap V(T)$ with respect to \prec . Let $U = \{x(u) | u \in Y\}$. For a vertex $u \in Y$ and $v \in N_G(u) \cap V(T) \setminus \{x(u)\}$, let $y(u, v)$ be the parent of v in T . Let $W = \{y(u, v) | u \in Y, v \in N_G(u) \cap V(T), v \neq x(u)\}$ and $W' = W \setminus (S \cup D \cup U)$. In particular, for the example on Fig. 2, we have that $x(u_1) = v_5$, $x(u_2) = v_{10}$, $x(u_3) = v_{11}$ and $U = \{v_5, v_{10}, v_{11}\}$. Respectively, $x(u_1, v_6) = v_4$, $x(u_1, v_{16}) = x(u_2, v_{16}) = v_{14}$, $x(u_3, v_{20}) = v_{19}$, $W = \{v_4, v_{14}, v_{19}\}$ and $W' = \{v_{14}\}$.

Let $F = G[V(T) \cup Y]$.

Claim 1. *Set $F' = F - W'$ is connected.*

Proof of the claim. Since all leaves of T including z are in S , we have that $z \in V(F')$. To prove the claim, we show that for each vertex $v \in V(F')$, there is a (v, z) -path in F' . Every vertex $u \in Y$ has a neighbor $x(u)$ in F' . Hence, it is sufficient to prove the existence of (v, z) -paths for $v \in V(T) \setminus W'$. The proof is by induction on the number of v with respect to \prec . The first vertex is z , and if $v = z$, then we have a trivial (z, v) -path. Assume that $v \neq z$. Let w be the parent of v in T . If $w \in V(F')$, then $w \prec v$ and, by the inductive hypothesis, there is a (z, w) -path in F' and it implies the existence of a (z, v) -path. Suppose that $w \notin V(F')$, i.e., $w \in W'$. Since $d_T(w) = 2$, there is $u \in Y$ such that $w = y(u, v)$. We have that $x(u) \prec v$ and there is a $(z, x(u))$ -path in F' by the inductive hypothesis. It remains to observe that because $x(u)u, uv \in E(F')$, F' has a (z, v) -path as well. This concludes the proof to the claim. \square

Denote by R the set of the children of the vertices of $D \cup S$ in T ; in particular, $R = \{v_2, v_5, v_6, v_{11}, v_{13}, v_{14}, v_{20}, v_{21}\}$ for the example on Fig. 2. Observe that $|N_G(Y) \cap V(T)| \leq |D \cup S| + |R| + |U| + |W'|$. Recall that $|V(F)| \leq \ell + r$. Because F' is connected and $S \subseteq V(F')$, $|V(F')| \geq \ell$. Hence, $|W'| \leq r$. Let $q_1 = |L|$, $q_2 = |V(T) \setminus (L \cup D)|$ and $q_3 = |D|$. We have that $q_1 + 2q_2 + 3q_3 \leq 2|E(T)| = 2(q_1 + q_2 + q_3 - 1)$. Then $q_3 \leq q_1 - 2$ and $|D \cup S| \leq 2|S| - 2 = 2p - 2$, because $L \subseteq S$. Let T' be the tree obtained from T by consequent dissolving all the vertices of degree 2 that are not in S . Then $|R| \leq |E(T')| \leq 2|S| - 3 = 2p - 3$. Since $|V(T)| \geq \ell$, $|U| \leq |Y| \leq r$. We obtain that $|N_G(Y) \cap V(T)| \leq |D \cup S| + |R| + |U| + |W'| \leq 2p - 2 + 2p - 3 + r + r = 4p + 2r - 5$. \square

Now we are ready to prove the main result of the section.

Theorem 7. SECLUDED STEINER TREE can be solved in time $2^{\mathcal{O}(p+r)} \cdot nm \cdot \log W$ by a true-biased Monte-Carlo algorithm and in time $2^{\mathcal{O}(p+r)} \cdot nm \log n \cdot \log W$ by a deterministic algorithm for graphs with n vertices and m edges, where $r = k - \ell$ and ℓ is the size of a Steiner tree for S and W is the maximum value of ω on an input graph G .

Proof. We construct an FPT-algorithm for SECLUDED STEINER TREE parameterized by $p + r$. The algorithm is based on the random separation techniques introduced by Cai, Chan, and Chan [8] (see also [1]). We first describe a randomized algorithm and then explain how it can be derandomized.

Let $\mathcal{I} = (G, \omega, S, k, C)$ be an instance of SECLUDED STEINER TREE, ℓ be the size of a Steiner tree for $S = \{s_1, \dots, s_p\}$ and $r = k - \ell$. Without loss of generality we assume that $p \geq 2$ and $r \geq 1$ as for $p = 1$ or $r = 0$, the problem is trivial. We also can assume that G is connected.

Before we give a formal description of the algorithm, we briefly sketch the main ideas. Let \mathcal{I} be a yes-instance. Then G has a connected induced

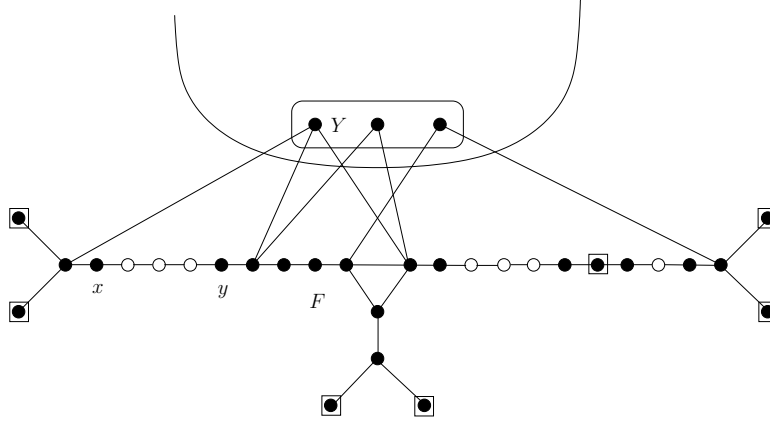


Figure 3: The structure of a solution; the terminals are shown by squares, the marked vertices of F are black and non-marked vertices are white.

subgraph F such that $S \subseteq V(F)$, $|N_G[V(F)]| \leq k$ and $\omega(N_G[V(F)]) \leq C$. Consider $Y = N_G[V(F)]$. Notice that Y separates F from the remaining part of the graph and $|Y| \leq r$ (see Fig. 3). By Lemmas 1 and 2, we obtain that the number vertices of F that are terminals, have degree at least three or adjacent to some vertices Y is $O(r + p)$. Moreover, the number of vertices of F that are adjacent in F to the aforementioned vertices is also $O(p + r)$. Let us call all these vertices of F *marked*. If we color the vertices of G independently and uniformly at random by two colors *red* and *blue*, then with the sufficiently high probability $(1/2)^{O(r+p)}$ that depends only on r and p , the marked vertices of F are colored red and the vertices of Y are colored blue. Suppose that the considered random coloring achieves the desired coloring of the marked vertices and the vertices of Y . We can still have non-marked vertices of F that are blue. Notice that non-marked vertices form induced paths in G such that only the end-vertices of each path have marked neighbors in G and these neighbors are pendant marked vertices of F (e.g, the vertices x and y in the example in Fig. 3). Then the non-marked blue vertices can be recognized and recolored red. This way, we obtain F colored red that is separated from the remaining part of the graph by blue vertices.

Description of the algorithm. In each iteration of the algorithm we color the vertices of G independently and uniformly at random by two colors. In other words, we partition $V(G)$ into two sets R and B . We say that the vertices of R are *red*, and the vertices of B are *blue*. Our algorithm can recolor some blue vertices red, i.e., the sets R and B can be modified. Our aim is to find a connected subgraph T of G with $S \subseteq V(T)$ such that $|N_G[V(T)]| \leq k$, $\omega(N_G[V(T)]) \leq C$ and $V(T) \subseteq R$.

Step 1. If $G[R]$ has a component H such that $S \subseteq V(H)$, then find a

spanning tree T of H . If $|N_G[V(T)]| \leq k$ and $\omega(N_G[V(T)]) \leq C$, then return T and stop; otherwise, return that \mathcal{I} is no-instance and stop.

Step 2. If there is $s_i \in S$ such that $s_i \notin R$ or $N_G(s_i) \cap R = \emptyset$, then return that \mathcal{I} is no-instance and stop.

Step 3. Find a component H of $G[R]$ with $s_1 \in V(H)$. If there is a pendant vertex $u \notin S$ of H that is adjacent in G to a unique vertex $v \in B$, then find a component of $G[B]$ that contains v , recolor its vertices red and then return to Step 1. Otherwise, return that (G, S, k) is no-instance and stop.

We repeat at most $2^{\mathcal{O}(r+p)}$ iterations. If on some iteration we obtain a yes-answer, then we return it and the corresponding solution. Otherwise, if on every iteration we get a no-answer, we return a no-answer.

Correctness of the algorithm. It is straightforward to see that if this algorithm returns a tree T in G with $|N_G[V(T)]| \leq k$ and $\omega(N_G[V(T)]) \leq C$, then we have a solution for the considered instance of SECLUDED STEINER TREE. We show that if \mathcal{I} is a yes-instance, then there is a positive constant α that does not depend on n and r such that the algorithm finds a tree T in G with $|N_G[V(T)]| \leq k$ and $\omega(N_G[V(T)]) \leq C$ with probability at least α after $2^{\mathcal{O}(p+r)}$ executions of this algorithm for random colorings.

Suppose that \mathcal{I} is a yes-instance. Then there is a tree T in G such that $S \subseteq V(T)$, $|N_G[V(T)]| \leq k$ and $\omega(N_G[V(T)]) \leq C$. Without loss of generality we assume that T is inclusion minimal. Let $F = G[V(T)]$, $X = \{v \in V(F) | d_F(v) \geq 3\} \cup S$, $X' = N_F(X)$, $Y = N_G(V(T))$ and $Y' = N_G(Y) \cap V(T)$. For each $v \in Y' \setminus S$, we arbitrarily select two distinct neighbors $z_1(v)$ and $z_2(v)$ in T . Because the leaves of T are in S , we have that v is not a leaf and thus has at least two neighbors. Let $Z = \{z_i(v) | v \in Y' \setminus S, i = 1, 2\}$. Let $W = X \cup X' \cup Y' \cup Z$.

By Lemma 1, $|X| \leq 4p - 6$ and $|X'| \leq 4p - 6$. By Lemma 2, $|Y'| \leq 4p + 2r - 5$ and, therefore, $|Z| \leq 8p + 4r - 10$. Because $|V(T)| \geq \ell$ and $|N_G[V(T)]| \leq \ell + r$, we have that $|Y| \leq r$. Hence $|W| \leq |X| + |X'| + |Y| + |Y'| + |Z| \leq 4p - 6 + 4p - 6 + r + 4p + 2r - 5 + 8p + 4r - 10 = 20p + 7r - 27$. Let $N = 20p + 7r - 27$. Then with probability at least 2^{-N} , the vertices of Y are colored blue and the vertices of $X \cup X' \cup Y' \cup Z$ are colored red, i.e., $W \cap V(T) \subseteq R$ and $W \setminus V(T) \subseteq B$. The probability that for a random coloring, the vertices of W are colored incorrectly, i.e., $W \cap V(T) \cap B \neq \emptyset$ or $(W \setminus V(T)) \cap R \neq \emptyset$, is at most $1 - 2^{-N}$. Hence, if we consider 2^N random colorings, then the probability that the vertices of W are colored incorrectly for all the colorings is at most $(1 - 2^{-N})^{2^N}$, and with probability at least $1 - (1 - 2^{-N})^{2^N}$ for at least one coloring we will have $W \cap V(T) \subseteq R$ and $W \setminus V(T) \subseteq B$. Since $(1 - 2^{-N})^{2^N} \leq 1/e$, we have that $1 - (1 - 2^{-N})^{2^N} \geq 1 - 1/e$. Thus if \mathcal{I} is a yes-instance, after 2^N random colorings of G , we have that at least one of the colorings is successful with a constant success probability $\alpha = 1 - 1/e$.

Assume that for a random red-blue coloring of G , $W \cap V(T) \subseteq R$ and $W \setminus V(T) \subseteq B$. We show that in this case the algorithm finds a tree T' with $S \subseteq V(T') \subseteq V(T)$. Clearly, $|N_G[V(T')]| \leq |N_G[V(T)]| \leq k$ and $\omega(N_G[V(T')]) \leq \omega(N_G[V(T)]) \leq C$ in this case.

We claim that for every connected component H of $G[R]$, either $V(H) \subseteq V(T)$ or $V(H) \cap V(T) = \emptyset$. To obtain a contradiction, assume that there are $u, v \in V(H)$ such that $u \in V(T)$ and $v \notin V(T)$. Indeed, H is connected, and thus contains an (u, v) path P . Since P goes from $V(T)$ to $v \notin V(T)$, path P should contain a vertex $w \in N_G(V(T)) = Y$. But w is colored blue, which is a contradiction to the assumption that P is in the red component H . By the same arguments, for any component H of $G[B]$, either $V(H) \subseteq V(T)$ or $V(H) \cap V(T) = \emptyset$.

We consider Steps 1–3 of the algorithm and show their correctness.

Suppose that $G[R]$ has a component H such that $S \subseteq V(H)$. Because $S \subseteq W \cap V(T) \subseteq R$, $V(H) \cap V(T) \neq \emptyset$ and, therefore, $V(H) \subseteq V(T)$. Then for every spanning tree T' of H , $S \subseteq V(T')$ and $N_G[V(T')] \subseteq N_G[V(T)]$. Therefore, $|N_G[V(T')]| \leq |N_G[V(T)]| \leq k$ and $\omega(N_G[V(T')]) \leq \omega(N_G[V(T)]) \leq C$. Hence, if a component of $G[R]$ contains S , then we find a solution. This concludes the proof of the correctness of the first step.

Let us assume that the algorithm does not stop at Step 1. For the right coloring, because $S \subseteq X$ and $N_F(S) \subseteq X'$, for every $s_i \in S$, we have that $s_i \in R$. Moreover, because $p \geq 2$, at least one neighbor of s_i in G is in R . Thus the only reason why the algorithm stops at Step 2 is due to the wrong coloring. Consider the case when the algorithm does not stop after Step 2.

Suppose that H is a component of $G[R]$ with $s_1 \in V(H)$. Because the algorithm did not stop in Step 2, such a component H exists and has at least 2 vertices. Recall that $V(H) \subseteq V(T)$. Because we proceed in Step 1, we conclude that $S \setminus V(H) \neq \emptyset$. Then there is a vertex $u \in V(H)$ which has a neighbor v in T such that $v \in B$. If $u \in S$, then $v \in X'$, but this contradicts the assumption $X' \subseteq R$. Hence, $u \notin S$. Suppose that $d_H(u) \geq 2$. In this case $d_F(u) \geq 3$ and $v \in X'$; a contradiction. Therefore, u is a pendant vertex of H .

Let $u \notin S$ be an arbitrary pendant vertex of H . If u has no neighbors in B , then u is a leaf of T that does not belong to S but this contradicts the inclusion minimality of T . Assume that u is adjacent to at least two distinct vertices of B . Because T is an inclusion minimal tree spanning S , vertex u has at least two neighbors in T and u has a neighbor $v \in B$ in T . Let $w \in (N_G(u) \cap B) \setminus \{v\}$. If $w \in V(T)$, then $d_F(u) \geq 3$ and, therefore, $u \in X$ and $v, w \in X'$; a contradiction with $X' \subseteq R$. Hence, $w \notin V(T)$. Moreover, v is the unique neighbor of u in T that belongs to B . Then $w \in Y$ and $v \in \{z_1(u), z_2(u)\}$; a contradiction with $Z \subseteq R$. We obtain that u is adjacent in G to a unique vertex $v \in B$. Let H' be the component of $G[B]$ that contains v . Since T is an inclusion minimal tree that spans S , u has at least two neighbors in T . It implies that $v \in V(T)$, therefore $V(H') \subseteq V(T)$.

We recolor the vertices of H' red in Step 3. For the new coloring the vertices of Y are blue and the vertices of $W \setminus Y$ are red. Therefore, we keep the crucial property of the considered coloring but we increase the size of the component of $G[R]$ containing s_1 .

To conclude the correctness proof, it remains to observe that in Step 3 we increase the number of vertices in the component of $G[R]$ that contains s_1 . Hence, after at most n repeats of Steps 1-3, we obtain a component in $G[R]$ that includes S and return a solution in Step 1.

It is straightforward to verify that each of Steps 1-3 can be done in time $\mathcal{O}(m \log W)$. Because the number of iterations is at most n , we obtain that the total running time is $2^{\mathcal{O}(p+r)} \cdot nm \log W$.

This algorithm can be derandomized by standard techniques (see [1, 8]). The random colorings can be replaced by the colorings induced by *universal sets*. Let n and q be positive integers, $q \leq n$. An (n, q) -*universal set* is a collection of binary vectors of length n such that for each index subset of size q , each of the 2^q possible combinations of values appears in some vector of the set. It is known that an (n, q) -universal set can be constructed in FPT-time with the parameter q . The best construction is due to Naor, Schulman and Srinivasan [24]. They obtained an (n, q) -universal set of size $2^q \cdot q^{\mathcal{O}(\log q)} \log n$, and proved that the elements of the sets can be listed in time that is linear in the size of the set. In our case n is the number of vertices of G and $q = 20p + 7r - 27$. \square

We complement Theorem 7 by showing that it is unlikely that SECLUDED STEINER TREE is FPT if parameterized by r only. To show it, we use the standard reduction from the SET COVER problem (see, e.g., [23]). Notice that we prove that SECLUDED STEINER TREE is co-W[1]-hard, i.e., we show that it is W[1]-hard to decide whether we have a no-answer.

Theorem 8. SECLUDED STEINER TREE *without costs is co-W[1]-hard when parameterized by r , where $r = k - \ell$ and ℓ is the size of a Steiner tree for S .*

Proof. Recall that the SET COVER problem for a set U , subsets $X_1, \dots, X_m \subseteq U$ and a positive integer k , asks whether there are $k' \leq k$ sets $X_{i_1}, \dots, X_{i_{k'}}$ for $i_1, \dots, i_{k'} \in \{1, \dots, m\}$ that cover U , i.e., $U \subseteq \cup_{j=1}^{k'} X_{i_j}$. As it was observed in [21]¹, SET COVER is W[1]-hard when parameterized by $p = m - k$. To prove the theorem, we reduce this parameterized variant of SET COVER.

Let (U, X_1, \dots, X_m, k) be an instance of SET COVER. Let $U = \{u_1, \dots, u_n\}$. We construct the bipartite graph G as follows.

- i) Construct m vertices x_1, \dots, x_m and n vertices u_1, \dots, u_n .
- ii) For $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$, construct an edge $x_i u_j$ if $u_j \in X_i$.

¹Gutin et al. prove in [21] the statement for the dual HITTING SET problem.

iii) Construct a vertex y and join it with x_1, \dots, x_m by edges.

Let $S = \{y, u_1, \dots, u_n\}$ and $r = m - k - 1$.

Suppose that (U, X_1, \dots, X_m, k) is a yes-instance of SET COVER and assume that $X_{i_1}, \dots, X_{i_{k'}}$ cover U . Then $F = G[S \cup \{x_{i_1}, \dots, x_{i_{k'}}\}]$ is a connected subgraph of G and $S \subseteq V(F)$. Clearly, $|V(F)| \leq n + k + 1$. Let T be a Steiner tree for the set of terminals S . We have that $\ell = |V(T)| \leq |V(F)| \leq n + k + 1$. Notice that for any connected subgraph T' of G such that $S \subseteq V(T')$, $N_G[V(T')] = V(G)$. We have that for any connected subgraph T' of G with $S \subseteq V(T')$, $|N_G[V(T')]| = n + m + 1 > (n + k + 1) + (m - k - 1) \geq \ell + r$. Therefore, $(G, S, \ell + r)$ is a no-instance of SECLUDED STEINER TREE without costs.

Assume now that $(G, S, \ell + r)$ is a no-instance of SECLUDED STEINER TREE without costs. Let T be a Steiner tree for the set of terminals S . Because for any connected subgraph T' of G such that $S \subseteq V(T')$, $N_G[V(T')] = V(G)$, and because $(G, S, \ell + r)$ is a no-instance, $\ell = |V(T)| < |V(G)| - r = (n + m + 1) - (m - k - 1) = n + k + 2$. Let $\{x_{i_1}, \dots, x_{i_{k'}}\} = \{x_1, \dots, x_k\} \cap V(T)$. Since $|V(T)| \leq n + k + 1$, we obtain that $k' \leq k$. It remains to note that $X_{i_1}, \dots, X_{i_{k'}}$ cover U and, therefore, (U, X_1, \dots, X_m, k) is a yes-instance of SET COVER. \square

5 Structural parameterizations of Secluded Steiner Tree

In this section we obtain kernelization results for different structural parameterizations of the secluded connectivity problems. We consider parameterizations by the exposure size, the treewidth, maximum degree and the size of a vertex cover and feedback vertex set of an input graph.

As we consider parameterizations by the treewidth, the size of a vertex cover and feedback vertex set of an input graph, it is useful to recall the relations between these parameters. The following observation is a folklore knowledge.

Observation 1. *For any graph G , $\text{tw}(G) \leq \text{fvs}(G) + 1$ and $\text{fvs}(G) \leq \text{vc}(G)$.*

It implies that if a graph problem parameterized by the treewidth of an input graph is FPT or has a polynomial kernel, the same holds for the case when the problem is parameterized by the size of a feedback vertex set or a vertex cover. Similarly, if a problem is FPT or has a polynomial kernel when parameterized by the size of a feedback vertex set, it is FPT or has a polynomial kernel when parameterized by the size of a vertex cover. In the opposite direction, any hardness result for a problem parameterized by the size of a vertex cover implies the same hardness result for the parameterization by the treewidth or the size of a feedback vertex set.

We show that under reasonable complexity assumptions SECLUDED PATH without costs has no polynomial kernel when parameterized by $k + t + \Delta$, where t is the treewidth and Δ is the maximum degree of an input graph.

Theorem 9. SECLUDED PATH *without costs* on graphs of treewidth at most t and maximum degree at most Δ admits no polynomial kernel unless $\text{NP} \subseteq \text{co-NP} / \text{poly}$ when parameterized by $k + t + \Delta$.

Proof. We construct an OR-cross-composition of SECLUDED PATH without costs to the parameterized version of SECLUDED PATH. Recall that SECLUDED PATH without costs was shown to be NP-complete by Chechik et al. [9, 10]. We assume that two instances $(G, \{s_1, s_2\}, k)$ and $(G', \{s'_1, s'_2\}, k')$ of SECLUDED PATH without costs are equivalent if $|V(G)| = |V(G')|$ and $k = k'$. Let $(G_i, \{s_1^i, s_2^i\}, k)$ for $i \in \{1, \dots, p\}$ be equivalent instances of SECLUDED PATH, $|V(G_i)| = n \geq 4$. Without loss of generality, we assume that $p = 2^q$ for a positive integer q ; otherwise, we add minimum number of copies of $(G_1, \{s_1^1, s_2^1\}, k)$ to achieve this property. We construct the graph G as follows.

- i) Construct disjoint copies of G_1, \dots, G_p .
- ii) Construct a rooted binary tree T_1 of height q , denote the root by s_1 and identify $t = 2^q$ leaves of the tree with the vertices of s_1^1, \dots, s_1^p of G_1, \dots, G_p .
- iii) Construct a rooted binary tree T_2 of height q , denote the root by s_2 and identify $t = 2^q$ leaves of the tree with the vertices of s_2^1, \dots, s_2^p of G_1, \dots, G_p .

We set $k' = k + 4q$ and consider the instance $(G, \{s_1, s_2\}, k')$ of SECLUDED PATH. Notice that $\text{tw}(G_i) \leq n - 1$ and $\Delta(G_i) \leq n - 1$ for $i \in \{1, \dots, p\}$. Clearly, $\Delta(G) \leq n$, and it can be seen that $\text{tw}(G) \leq n - 1$ as we can construct a tree decomposition of G whose bags are either the set of vertices of G_1, \dots, G_p or 4-vertex bags composed by the end vertices of two symmetric edges of T_1 and T_2 .

We claim that G has an (s_1, s_2) -path P with $|N_G[V(P)]| \leq k'$ if and only if there is $i \in \{1, \dots, p\}$ such that G_i has an (s_1^i, s_2^i) -path P_i with $|N_{G_i}[V(P_i)]| \leq k$.

Let P be an (s_1, s_2) -path P in G with $|N_G[V(P)]| \leq k'$. Consider the first vertex u of P starting from s_1 that is a leaf of T_1 . Clearly, $u = s_1^i$ for some $i \in \{1, \dots, p\}$. Notice that P contains s_2^i by the construction of G and that the (s_1^i, s_2^i) -subpath P_i of P is an (s_1^i, s_2^i) -path in G_i . It remains to observe that $k' \geq |N_G[V(P)]| \geq 4q + |N_{G_i}[V(P_i)]|$ and, therefore, $|N_{G_i}[V(P_i)]| \leq k$.

Suppose that G_i has an (s_1^i, s_2^i) -path P_i with $|N_{G_i}[V(P_i)]| \leq k$ for some $i \in \{1, \dots, p\}$. Let P' be the unique (s_1, s_1^i) -path in T_1 and let P'' be the unique (s_2^i, s_2) -path in T_2 . We have that for the (s_1, s_2) -path P in G obtained

by the concatenation of P' , P_i in the copy of G_i and P'' , $|N_G[V(P)]| \leq k + 4q = k'$. \square

Observe that Theorem 9 immediately implies that SECLUDED PATH without costs has no polynomial kernel unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$ when parameterized only by k or the treewidth. The next natural question is if parameterization by a stronger parameter can lead to a polynomial kernel. The following theorem provides lower bounds for parameterization by the minimum size of a vertex cover.

Theorem 10. SECLUDED PATH without costs on graphs with the vertex cover number at most w has no polynomial kernel unless $\text{NP} \subseteq \text{co-NP}/\text{poly}$ when parameterized by w .

Proof. The proof uses the cross-composition technique introduced by Bodlaender, Jansen and Kratsch [6]. We show that the 3-SATISFIABILITY problem OR-cross-composes into SECLUDED PATH without costs. Recall that 3-SATISFIABILITY asks for given boolean variables x_1, \dots, x_n and clauses C_1, \dots, C_m with 3 literals each, whether the formula $\phi = C_1 \wedge \dots \wedge C_m$ can be satisfied. It is well-known that 3-SATISFIABILITY is NP-complete [19]. We assume that two instances of 3-SATISFIABILITY are equivalent if they have the same number of variables and the same number of clauses.

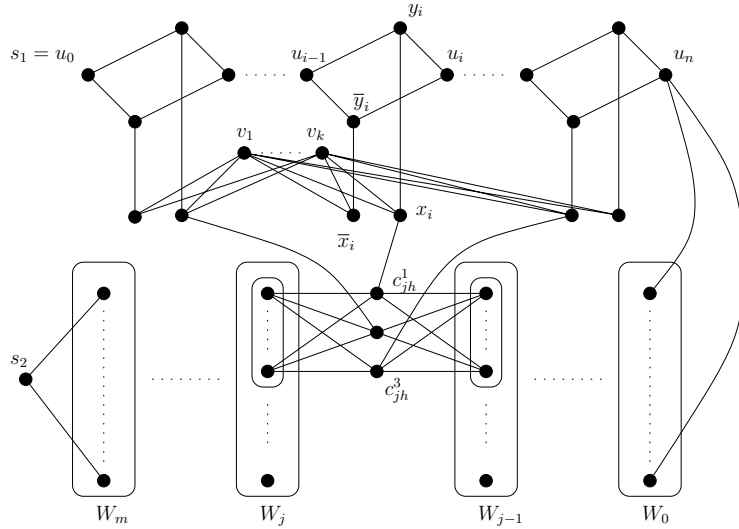


Figure 4: Construction of G .

Consider t equivalent instances of 3-SATISFIABILITY with the same boolean variables x_1, \dots, x_n and the sets of clauses $\mathcal{C}_i = \{C_1^i, \dots, C_m^i\}$ for $i \in \{1, \dots, t\}$. Without loss of generality we assume that $t = \binom{2q}{q}$ for a positive integer q ; otherwise, we add minimum number of copies of \mathcal{C}_1 to get this property. Notice that $\binom{2q}{q} = \Theta(4^q/\sqrt{\pi q})$ and $q = \mathcal{O}(\log t)$.

Let I_1, \dots, I_t be pairwise distinct subsets of $\{1, \dots, 2q\}$ of size q . Notice that each $i \in \{1, \dots, 2q\}$ is included exactly in $d = \binom{2q-1}{q-1}$ sets. Let $k = (q + 3d)m + 3q + 4n + 2$. We construct the graph G as follows (see Fig. 4).

- i) Construct $n + 1$ vertices u_0, \dots, u_n . Let $s_1 = u_0$.
- ii) For each $i \in \{1, \dots, n\}$, construct vertices $x_i, y_i, \bar{x}_i, \bar{y}_i$ and edges $u_{i-1}y_i, y_iu_i, y_ix_i$, and $u_{i-1}\bar{y}_i, \bar{y}_iu_i, \bar{y}_i\bar{x}_i$.
- iii) For each $j \in \{0, \dots, m\}$, construct a set of vertices $W_j = \{w_1^j, \dots, w_{2q}^j\}$.
- iv) Construct a vertex s_2 and edges $u_nw_1^0, \dots, u_nw_{2q}^0$ and $w_1^ms_2, \dots, w_{2q}^ms_2$.
- v) For each $j \in \{1, \dots, m\}$ and $h \in \{1, \dots, t\}$,
 - construct 3 vertices $c_{jh}^1, c_{jh}^2, c_{jh}^3$;
 - construct edges $c_{jh}^1w_r^{j-1}, c_{jh}^2w_r^{j-1}, c_{jh}^3w_r^{j-1}$ and $c_{jh}^1w_r^j, c_{jh}^2w_r^j, c_{jh}^3w_r^j$ for all $r \in I_h$;
 - consider the clause $C_j^h = (z_1 \vee z_2 \vee z_3)$ and for $l \in \{1, 2, 3\}$, construct an edge $c_{jh}^l x_i$ if $z_l = x_i$ for some $i \in \{1, \dots, n\}$ and construct an edge $c_{jh}^l \bar{x}_i$ if $z_l = \bar{x}_i$.
- vi) Construct k vertices v_1, \dots, v_k and edges x_iv_l, \bar{x}_iv_l for $i \in \{1, \dots, n\}$ and $l \in \{1, \dots, k\}$.

Observe that the set of vertices

$$X = (\cup_{i=1}^n \{x_i, y_i, \bar{x}_i, \bar{y}_i\}) \cup (\cup_{j=0}^m W_j)$$

is a vertex cover in G of size $4n + 2q(m + 1) = \mathcal{O}(n + m \log t)$.

We show that G has an (s_1, s_2) -path P with $|N_G[V(P)]| \leq k$ if and only if there is $h \in \{1, \dots, t\}$ such that x_1, \dots, x_n have a truth assignment satisfying all the clauses of \mathcal{C}_h .

Suppose that there is a truth assignment to x_1, \dots, x_n satisfying all the clauses of \mathcal{C}_h . First, we construct the (s_1, u_n) -path P' by the concatenation of the following paths: for each $i \in \{1, \dots, n\}$, we take the path $u_{i-1}y_iu_i$ if $x_i = \text{true}$ in the assignment and we take $u_{i-1}\bar{y}_iu_i$ if $x_i = \text{false}$. Let $r \in I_h$. We construct the (w_r^0, w_r^m) -path P'' by concatenating $w_r^{j-1}c_{jh}^{l_j}w_r^j$ for $j \in \{1, \dots, m\}$ where $l_j \in \{1, 2, 3\}$ is chosen as follows. Each clause $C_j^h = z_1 \vee z_2 \vee z_3 = \text{true}$ for the assignment, i.e., $z_l = \text{true}$ for some $l \in \{1, 2, 3\}$; we set $l_j = l$. Finally, we set $P = P' + u_nw_r^0 + P'' + w_r^ms_2$. It is straightforward to verify that $|N_G[V(P)]| = k$.

Suppose now that there is an (s_1, s_2) -path in G with $|N_G[V(P)]| \leq k$. We assume that P is an induced path. Observe that $x_i, \bar{x}_i \notin V(P)$ for

$i \in \{1, \dots, n\}$, because $d_G(x_i), d_G(\bar{x}_i) > k$. Therefore, P has an (s_1, u_n) -subpath P' such that $u_0, \dots, u_n \in V(P')$ and for each $i \in \{1, \dots, n\}$, either $y_i \in V(P')$ or $\bar{y}_i \in V(P')$. We set the variable $x_i = \text{true}$ if $y_i \in V(P')$ and $x_i = \text{false}$ otherwise. We show that this truth assignment satisfies all the clauses of some \mathcal{C}_h .

Observe that $|N_G[V(P')]| = 4n + 2q + 1$. Clearly, $s_2 \in V(P)$. Notice also that P has at least one vertex in each W_j for $j \in \{0, \dots, m\}$, and for each $j \in \{1, \dots, m\}$, at least one vertex among the vertices c_{jh}^l for $h \in \{1, \dots, t\}$ and $l \in \{1, 2, 3\}$ is in P . For each $j \in \{1, \dots, m\}$, any two vertices $w_r^{j-1} \in W_{j-1}$ and $w_{r'}^j \in W_j$ have at least $3d$ neighbors among the vertices c_{jf}^l for $f \in \{1, \dots, t\}$ and $l \in \{1, 2, 3\}$. Moreover, if $r \neq r'$, they have at least $3d+6$ such neighbors, because there are two subsets $I, I' \subseteq \{1, \dots, 2q\}$ of size q such that $r \in I \setminus I'$ and $r' \in I' \setminus I$. For each $j \in \{1, \dots, m-1\}$, any two vertices c_{jh}^l and $c_{j+1 h'}^{l'}$ for $h, h' \in \{1, \dots, t\}$ and $l, l' \in \{1, 2, 3\}$ have at least q neighbors in W_j . Moreover, if $h \neq h'$, they have at least $q+2$ such neighbors, because $|I_h \cup I_{h'}| \geq q+2$. Taking into account that $d_G(s_2) = 2q$, we obtain that

$$k \geq |N_G[V(P)]| \geq |N_G[V(P')]| + 3dm + q(m-1) + 2q + 1 = k.$$

It implies that P has exactly one vertex in each W_j for $j \in \{0, \dots, m\}$, and for each $j \in \{1, \dots, m\}$, exactly one vertex among the vertices c_{jh}^l for $h \in \{1, \dots, t\}$ and $l \in \{1, 2, 3\}$ is in P . Moreover, there is $r \in \{1, \dots, 2q\}$ and $h \in \{1, \dots, t\}$ such that $w_r^j \in V(P)$ and $c_{jh}^l \in V(P)$ for $j \in \{0, \dots, m\}$ and $l_j \in \{1, 2, 3\}$. We claim that all the clauses of \mathcal{C}_r are satisfied. Otherwise, if there is a clause $C_j^r = (z_1 \vee z_2 \vee z_3)$ that is not satisfied, then the neighbors of $c_{jh}^1, c_{jh}^2, c_{jh}^3$ among the vertices x_i, \bar{x}_i for $i \in \{1, \dots, n\}$ are not in $N_G[V(P')]$. It immediately implies that $|N_G[V(P)]| > k$; a contradiction. \square

However, if we consider parameterization by the exposure size and the size of a feedback vertex set, then we obtain the following theorem.

Theorem 11. *The SECLUDED STEINER TREE problem admits a kernel such that the output graph has $\mathcal{O}(k^3 r)$ vertices on graphs with a feedback vertex set of size at most r .*

Proof. Let (G, ω, S, k, C) be an instance of SECLUDED STEINER TREE.

Becker and Geiger proved in [3] (see also [2]) that there is a factor two polynomial approximation for a minimum feedback vertex set. Let X be a feedback vertex set of G of size at most $2\text{fvs}(G) \leq 2r$. Let $F = G - X$. Clearly, F is a forest. Let $W = \{v \in X \mid d_G(v) \geq k\}$ and $Y = X \setminus W$. Let also $Z = (N_G(Y) \cup S) \cap V(F)$. See Fig. 5 for an example. Notice that $|Z| \leq |Y|(k-1) + k \leq 2r(k-1) + k$.

We say that $v \in V(F)$ is a *leaf* if $d_F(v) = 1$. We call a vertex $u \in V(F)$ a *sub-leaf* of F if it has at least one leaf neighbor and at most one nonleaf neighbor.

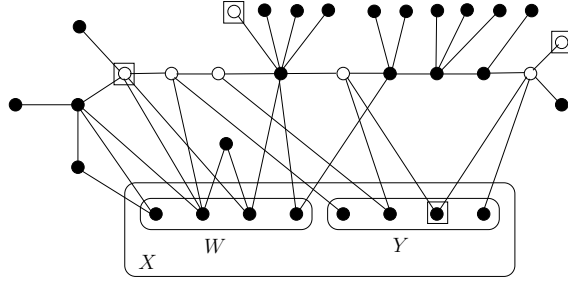


Figure 5: An example of G with a feedback vertex set X ; the terminals are shown by squares, the vertices of $Z = (N_G(Y) \cup S) \cap V(F)$ are white and the other vertices are black.

Observe that if T is a solution for (G, ω, S, k, C) , then $V(T) \cap W = \emptyset$. To construct a kernel, we apply reduction rules that delete vertices of F . Such deletions can modify degrees of the vertices of W , but we need to forbid the usage of the vertices of W in solutions. To do it, we apply the following rule (see Fig. 6 for an example).

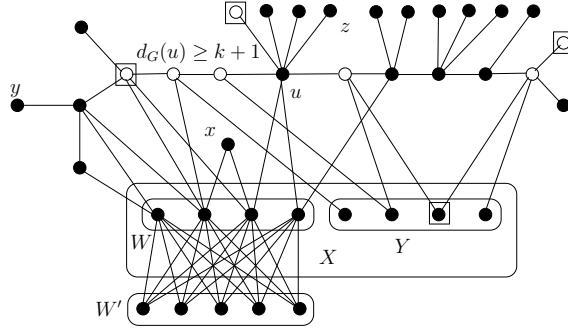


Figure 6: The application of Rules 1–4 for the graph shown in Fig. 5.

Reduction Rule 1 (Padding). Construct a set W' of k vertices and make them adjacent to each vertex of W . Denote the obtained graph by G' . Set $\omega(v) = 1$ for $v \in W'$.

It is straightforward to see that (G, ω, S, k, C) is a yes-instance of SECLUDED STEINER TREE if and only if (G', ω, S, k, C) is a yes-instance. Observe that X is a feedback vertex set of G' , i.e., $\text{fvs}(G') \leq 2r$.

Reduction Rule 2 (Isolates removal). If there is $v \in V(F) \setminus Z$ which is an isolated vertex of F , then delete v .

In particular, the vertex x in Fig. 6 is deleted by the rule.

To see that the rule is *safe*, that is, it returns an equivalent instance of the problem, denote by G'' the graph obtained from G' by the application

of the rule for an isolated $v \in V(F) \setminus Z$. Assume that T is a solution for (G', ω, S, k, C) . Because v can be adjacent in G' only to some vertices of W , we have that $v \notin V(T)$. Hence, T is a solution for (G'', ω, S, k, C) . If T is a solution for the obtained instance, then $v \notin N_{G'}[V(T)]$, because $W \cap V(T) = \emptyset$ and v can be adjacent in G' only to vertices of W . Hence, T is a solution for the original instance.

Reduction Rule 3 (Leaf removal I). *If there is a leaf v of F that is adjacent to a subleaf u of F such that $u, v \in V(F) \setminus Z$ and u has at most one neighbor w in F with the one of the following properties: i) $w \in Z$ or ii) w a non-leaf of F , then delete v .*

For example, the vertex y in Fig. 6 is deleted by the rule.

To show that the rule is safe, denote by G'' the graph obtained from G' by the application of the rule for a leaf $v \in V(F) \setminus Z$ and a sub-leaf $u \in V(F) \setminus Z$. Suppose that T is a solution for (G', ω, S, k, C) . Assume without loss of generality that T is inclusion minimal. Notice that all the vertices of $N_F(u)$ except maybe one are adjacent in G only to u and some vertices of W and $N_G[u] \setminus W = N_F[u]$. Then $v \notin V(T)$, because $W \cap T = \emptyset$, and, moreover, $u \notin V(T)$. Hence, $N_{G'}[V(T)] = N_G[V(T)]$. We obtain that T is a solution for (G'', ω, S, k, C) . For the other direction, let T be a solution for the instance obtained by the application of the rule. Again, assume that T is inclusion minimal. Recall that u is a sub-leaf of F and it holds that u has at most one neighbor in F that is in Z or a non-leaf of F . Since $N_G[u] \setminus W = N_F[u]$, we obtain that $u \notin V(T)$. Therefore, T is a solution for (G', ω, S, k, C) .

Reduction Rule 4 (Leaf removal II). *If there is leaf v of F such that $v \in V(F) \setminus Z$ and $d_G(u) \geq k + 1$ for its unique neighbor u in F , then delete v .*

For example, the vertex z in Fig. 6 is deleted by the rule.

Again, denote by G'' the graph obtained from G' by the application of the rule for a leaf $v \in V(F) \setminus Z$. Let u be the neighbor of v in F . Suppose that T is a solution for (G', ω, S, k, C) . Because $d_G(u) \geq k + 1$, $u \notin V(T)$. Since v is adjacent in G to u and some vertices of W , $v \notin V(T)$. We conclude that T is a solution for (G'', ω, S, k, C) . Let now T be a solution for (G'', ω, S, k, C) . Notice that $u \notin V(T)$, because $d_{G'}(u) \geq k$. Then T is a solution for (G', ω, S, k, C) .

Reduction Rule 5 (Path removal). *If there are two vertices u and v of a component of F such that for the (u, v) -path P in F with the set of internal vertices $U = V(P) \setminus \{u, v\}$, the following holds:*

- i) the length of P is at least 3,*
- ii) $(N_F[U] \setminus \{u, v\}) \cap Z = \emptyset$,*

iii) $|N_G[V(P)]| \geq k + 1$,

iv) the vertices of $N_F(U) \setminus \{u, v\}$ are leaves of F ,

then delete the edges of $F[U]$ and the vertices of $N_F[U] \setminus (N_F[u] \cup N_F[v])$.

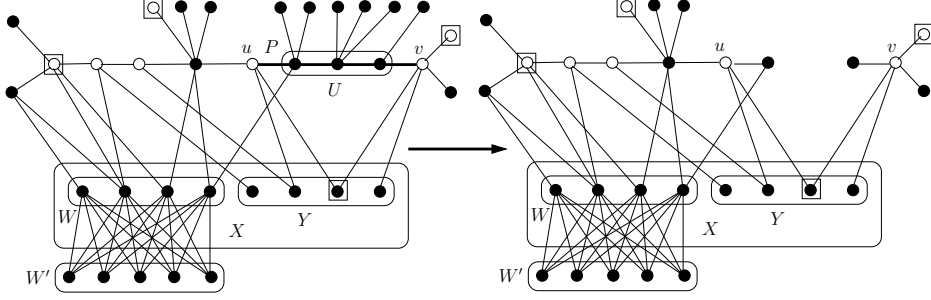


Figure 7: The application of Rule 5.

See Fig. 7 for an example. Let us remark, that in Rule 5, we need edge deletions only when the length of P is 3. When the length of P is more than 3, due to vertex deletions, edge deletion becomes obsolete.

Let G'' be the graph obtained from G' by the application of the rule for $u, v \in V(F)$ such that for the (u, v) -path P , the conditions i)–iv) hold. Suppose that T is an inclusion minimal solution for (G', ω, S, k, C) . For $x \in N_F[U] \setminus \{u, v\}$, any neighbor $y \notin V(F)$ is in W by ii) and iv). By the minimality of T and iii), $(N_F[U] \setminus \{u, v\}) \cap V(T) = \emptyset$, because, otherwise, $V(P) \subseteq V(T)$. Therefore, T is a solution for (G'', ω, S, k, C) . Let now T be an inclusion minimal solution for (G'', ω, S, k, C) . Notice that the neighbors u' and v' of u and v respectively that are vertices of P cannot be in T , because $|N_{G'}(u') \setminus W| = 1$ and $|N_{G'}(v') \setminus W| = 1$. It implies that T is a solution for (G', ω, S, k, C) .

We proved that Rules 2–5 are safe. We apply them exhaustively. To simplify notations, assume that these rules cannot be applied any more for (G', ω, S, k, C) and assume that F is the forest induced by $V(G') \setminus (X \cup W')$.

We claim that $|V(F)| = \mathcal{O}(k^3 r)$. Notice that if $Z = \emptyset$, then F is empty, because of Rules 2 and 3. Assume that $Z \neq \emptyset$. Let F' be the forest obtained from F by the removal of the leaves that are not in Z . Let

$$U = \{v \in V(F') \mid v \in Z \text{ or } d_{F'}(v) \neq 2\}.$$

Because of Rules 2 and 3, every leaf and every isolated vertex of F' is a vertex of Z . Hence, $|U| \leq 2|Z| - 2$. Let $u, v \in U$ be vertices of a component of F' such that the internal vertices of the (u, v) -path P have degrees 2 in F' and do not belong to U . Because of Rule 5, P has length at most $\max\{2, k - 1\}$, i.e., P has at most $\max\{1, k - 2\}$ internal vertices. This implies that $V(F') \setminus Z$ has at most $(2|Z| - 3) \cdot \max\{1, k - 2\}$ vertices of degree 2 if $|Z| \geq 2$ and no such

vertices if $|Z| = 1$. Therefore, $|V(F')| \leq 2|Z|(k+1)$. Since $|Z| \leq 2r(k-1)+k$, $|V(F')| \leq 4k(k+1)(r+1)$. Because of Rule 4, each vertex of F' is adjacent to at most k vertices of $V(F)$. Hence, $|V(F)| \leq 4k^2(k+1)(r+1)$.

Since $|X| \leq 2r$ and $|W'| = k$, we have that $|V(G')| = |X| + |W'| + |V(F)| = \mathcal{O}(k^3r)$. Hence, we obtain a kernel with $\mathcal{O}(k^3r)$ vertices.

Notice that none of the rules changes the parameter k but Rule 1 can increase the size of a feedback vertex set. Still, the obtained graph has a feedback vertex set of size at most $2r$.

It remains to observe that Rules 1–5 can be applied in polynomial time, which completes the proof. \square

Finally, let us remark that all reduction rules in the proof of the theorem do not change the costs of vertices. Therefore, we can also claim the $\mathcal{O}(k^3 \cdot \mathbf{fvs}(G))$ kernel for SECLUDED STEINER TREE without costs by assigning unit costs to the vertices of W' .

References

- [1] N. ALON, R. YUSTER, AND U. ZWICK, *Color-coding*, J. ACM, 42 (1995), pp. 844–856.
- [2] V. BAFNA, P. BERMAN, AND T. FUJITO, *A 2-approximation algorithm for the undirected feedback vertex set problem*, SIAM J. Discrete Math., 12 (1999), pp. 289–297.
- [3] A. BECKER AND D. GEIGER, *Optimization of Pearl’s method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem*, Artif. Intell., 83 (1996), pp. 167–188.
- [4] A. BJÖRKLUND, T. HUSFELDT, P. KASKI, AND M. KOIVISTO, *Fourier meets Möbius: fast subset convolution*, in Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11–13, 2007, ACM, 2007, pp. 67–74.
- [5] H. L. BODLAENDER, R. G. DOWNEY, M. R. FELLOWS, AND D. HERMELIN, *On problems without polynomial kernels*, J. Comput. Syst. Sci., 75 (2009), pp. 423–434.
- [6] H. L. BODLAENDER, B. M. P. JANSEN, AND S. KRATSCH, *Kernelization lower bounds by cross-composition*, SIAM J. Discrete Math., 28 (2014), pp. 277–305.
- [7] L. CAI, *Parameterized complexity of cardinality constrained optimization problems*, Comput. J., 51 (2008), pp. 102–121.

- [8] L. CAI, S. M. CHAN, AND S. O. CHAN, *Random separation: a new method for solving fixed-cardinality optimization problems*, in IWPEC, vol. 4169 of Lecture Notes in Computer Science, Springer, 2006, pp. 239–250.
- [9] S. CHECHIK, M. P. JOHNSON, M. PARTER, AND D. PELEG, *Secluded connectivity problems*, CoRR, abs/1212.6176 (2012).
- [10] S. CHECHIK, M. P. JOHNSON, M. PARTER, AND D. PELEG, *Secluded connectivity problems*, in Proceedings of the 21st Annual European Symposium Algorithms (ESA), vol. 8125 of Lecture Notes in Computer Science, Springer, 2013, pp. 301–312.
- [11] M. CYGAN, F. V. FOMIN, L. KOWALIK, D. LOKSHTANOV, D. MARX, M. PILIPCZUK, M. PILIPCZUK, AND S. SAURABH, *Parameterized algorithms*, Springer, 2015.
- [12] R. G. DOWNEY AND M. R. FELLOWS, *Fundamentals of Parameterized Complexity*, Texts in Computer Science, Springer, 2013.
- [13] S. E. DREYFUS AND R. A. WAGNER, *The Steiner problem in graphs*, Networks, 1 (1971), pp. 195–207.
- [14] M. R. FELLOWS, D. HERMELIN, F. A. ROSAMOND, AND S. VIALETTE, *On the parameterized complexity of multiple-interval graph problems*, Theor. Comput. Sci., 410 (2009), pp. 53–61.
- [15] F. V. FOMIN, P. A. GOLOVACH, N. KARPOV, AND A. S. KULIKOV, *Parameterized complexity of secluded connectivity problems*, in FSTTCS 2015, vol. 45 of LIPIcs, 2015, pp. 408–419.
- [16] F. V. FOMIN AND D. KRATSCH, *Exact exponential algorithms*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, 2010.
- [17] F. V. FOMIN AND Y. VILLANGER, *Treewidth computation and extremal combinatorics*, Combinatorica, 32 (2012), pp. 289–308.
- [18] J. GAO, Q. ZHAO, AND A. SWAMI, *The thinnest path problem for secure communications: A directed hypergraph approach*, in Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing, 2012, IEEE, 2012, pp. 847–852.
- [19] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
- [20] A. GILBERS, *Visibility domains and complexity*, PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, 2013.

- [21] G. GUTIN, M. JONES, AND A. YEO, *Kernels for below-upper-bound parameterizations of the hitting set and directed dominating set problems*, Theor. Comput. Sci., 412 (2011), pp. 5744–5751.
- [22] M. P. JOHNSON, O. LIU, AND G. RABANCA, *Secluded path via shortest path*, in SIROCCO 2014, vol. 8576 of Lecture Notes in Computer Science, Springer, 2014, pp. 108–120.
- [23] R. M. KARP, *Reducibility among combinatorial problems*, in Proceedings of a symposium on the Complexity of Computer Computations, The IBM Research Symposia Series, Plenum Press, New York, 1972, pp. 85–103.
- [24] M. NAOR, L. SCHULMAN, AND A. SRINIVASAN, *Splitters and near-optimal derandomization*, in 36th Annual Symposium on Foundations of Computer Science (FOCS 1995), IEEE, 1995, pp. 182–191.
- [25] J. NEDERLOF, *Fast polynomial-space algorithms using inclusion-exclusion*, Algorithmica, 65 (2013), pp. 868–884.
- [26] K. PIETRZAK, *On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems*, J. Comput. Syst. Sci., 67 (2003), pp. 757–771.