

The parameterized complexity of finding secluded solutions to some classical optimization problems on graphs[☆]

René van Bevern^{a,b,1}, Till Fluschnik^{c,2,*}, George B. Mertzios^{d,3}, Hendrik Molter^{c,4}, Manuel Sorge^{c,e,5}, Ondřej Suchý^{f,6}

^a *Novosibirsk State University, Novosibirsk, Russian Federation,*

^b *Sobolev Institute of Mathematics of the Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russian Federation*

^c *Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany,*

^d *School of Engineering and Computing Sciences, Durham University, Durham, UK,*

^e *Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Be'er Sheva, Israel*

^f *Faculty of Information Technology, Czech Technical University in Prague, Prague, Czech Republic*

Abstract

This work studies the parameterized complexity of finding *secluded* solutions to classical combinatorial optimization problems on graphs such as finding minimum s - t separators, feedback vertex sets, dominating sets, maximum independent sets, and vertex deletion problems for hereditary graph properties: Herein, one searches not only to minimize or maximize the size of the solution, but also to minimize the size of its neighborhood. This restriction has applications in secure routing and community detection.

Keywords:

Neighborhood, Feedback Vertex Set, Vertex Deletion, Separator, Dominating Set

1. Introduction

In many optimization problems on graphs, one searches for a minimum or maximum cardinality subset of vertices and edges satisfying certain properties, like a shortest s - t path, a maximum independent set, or a minimum dominating set. Chechik et al. [12] first studied the problem of finding *secluded* solutions, which additionally limit the *exposure* of the solution as measured by the size of the neighborhood. They motivate these problems by protecting sensitive information that is sent through a network and potentially intercepted by neighbors of its travel path. However, given that there are effective means of encrypting and signing

[☆]A preliminary version of this work appeared in the proceedings of IPEC 2016 [5]. This version provides full proof details and additionally shows that SMALL SECLUDED s - t -SEPARATOR is fixed-parameter tractable parameterized by the combination of the solution size and the open neighborhood size (Theorem 3.5).

*Corresponding author.

Email addresses: `rvb@nsu.ru` (René van Bevern), `till.fluschnik@tu-berlin.de` (Till Fluschnik), `george.mertzios@durham.ac.uk` (George B. Mertzios), `h.molter@tu-berlin.de` (Hendrik Molter), `sorge@post.bgu.ac.il` (Manuel Sorge), `ondrej.suchy@fit.cvut.cz` (Ondřej Suchý)

¹Results in Section 5 were obtained under support of the Russian Science Foundation, grant 16-11-10041. This study was initiated during a visit to TU Berlin jointly supported by the Russian Foundation for Basic Research, grant 16-31-60007 mol_a_dk, by the Ministry of Science and Higher Education of the Russian Federation under the 5-100 excellence programme, and by TU Berlin.

²Supported by the DFG, project DAMM (NI 369/13).

³Supported by the EPSRC project EP/P020372/1.

⁴Supported by the DFG, project DAPA (NI 369/12).

⁵Supported by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number 631163.11 and by the Israel Science Foundation (grant no. 551145/14).

⁶Supported by grant 17-20065S of the Czech Science Foundation.

sensitive information, the following application seems more realistic: a convoy travelling from a vertex s to a vertex t along an s - t path in a transportation network can potentially be attacked from roads incident to this path. Thus, one arrives at the problem of finding an s - t path with a small closed neighborhood. Another motivation for limiting the exposure of solutions is the search for segregated communities in social networks [31]. Here we search for dense subgraphs that are exposed to few neighbors in the rest of the graph. The constrained exposure models the concept of inter-cluster sparsity, which states that communities have weak connections to the rest of the network [26]. In addition to being a natural constraint in the above applications, restricting the exposure of the solution may also yield more efficient algorithms [28, 30, 31, 34].

Chechik et al. [12] and Fomin et al. [25] previously studied secluded paths and Steiner trees, respectively. Our aim in this paper is to study the classical and parameterized complexity of secluded variants of classical combinatorial optimization problems in graphs.

Following Chechik et al. [12], we measure the exposure of a solution S by the size of the closed neighborhood $N_G[S] = S \cup \bigcup_{v \in S} N_G(v)$ of S in the input graph G . Given a predicate $\Pi(G, S)$ that determines whether S is a solution for input graph G , we study the following general problem.

Problem (SECLUDED Π).

Input: A graph $G = (V, E)$ and an integer k .

Question: Is there a vertex subset $S \subseteq V$ such that S satisfies $\Pi(G, S)$ and $|N_G[S]| \leq k$?

In some cases, it may be necessary to control the size of the solution and its neighborhood independently: For example, when routing a convoy from s to t as above, we may simultaneously aim to minimize its travel time, that is, the number of vertices on its route, and to limit the exposure. Hence, another measure for the exposure of the solution is the size of the open neighborhood $N_G(S) = N_G[S] \setminus S$. We thus introduce and study the complexity of the following problem.

Problem (SMALL (LARGE) SECLUDED Π).

Input: A graph $G = (V, E)$ and two integers k, ℓ .

Question: Is there a vertex subset $S \subseteq V$ such that S satisfies $\Pi(G, S)$, $|S| \leq k$, and $|N_G(S)| \leq \ell$ (or $|S| \geq k$, and $|N_G(S)| \leq \ell$, respectively)?

Our contributions. We study SECLUDED Π and SMALL SECLUDED Π in the framework of parameterized complexity, a framework allowing for a fine-grained complexity analysis and for proving the effectivity of polynomial-time data reduction (we give formal definitions in Section 2): a problem is *fixed-parameter tractable* with respect to some parameter k if it can be solved in $f(k) \cdot n^{O(1)}$ time, where n is the input size. Thus, for small parameters k , fixed-parameter algorithms can potentially lead to efficient algorithms for NP-hard problems. In contrast, if a problem is W[1]-hard with respect to a parameter k , then it is presumably not fixed-parameter tractable with respect to that parameter. Our results are summarized in Table 1.

We analyze the impact of the parameter k on the complexity of SECLUDED Π and the impact of the parameters k and ℓ on the complexity of SMALL SECLUDED Π . The predicates $\Pi(G, S)$ that we study are

- s - t SEPARATOR,
- FEEDBACK VERTEX SET (FVS),
- \mathcal{F} -FREE VERTEX DELETION (\mathcal{F} -FVD) (for an arbitrary finite family \mathcal{F} of graphs), encompassing CLUSTER VERTEX DELETION, for example, and
- INDEPENDENT SET (IS).

Perhaps surprisingly, we find that SECLUDED s - t -SEPARATOR is polynomial-time solvable, whereas SMALL SECLUDED s - t SEPARATOR is NP-complete. The remaining secluded problem variants are NP-complete. For them, roughly speaking, we prove that fixed-parameter tractability results for Π parameterized by the solution size carry over to SECLUDED Π parameterized by the closed neighborhood size k . For SMALL SECLUDED Π parameterized by the open neighborhood size ℓ , however, we mostly obtain W[1]-hardness.

Problem	Complexity		Parameterized Complexity / Kernelization			
	<i>Secluded</i>	<i>Small Secl.</i>	<i>Secluded</i> k	<i>Small Secl.</i> k	ℓ	$k + \ell$
s - t Separator	P (Thm. 3.1)	NP-c. (Thm. 3.4)	P	W[1]-h. (Thm. 3.4)	W[1]-h. (Thm. 3.4)	FPT/noPK (Thm. 3.5/Thm. 3.8)
q -Dom. Set, $2p \leq q$	NP-c. (Thm. 4.3)	NP-c.*	W[2]-h. (Thm. 4.4)	\rightarrow	\rightarrow	W[2]-h. (Cor. 4.7)
q -Dom. Set, $2p > q$	NP-c. (Thm. 4.3)	NP-c.*	FPT/noPK (Cor. 4.9/Thm. 4.3)	W[2]-h.*	?	FPT/noPK (Cor. 4.7/Thm. 4.8)
\mathcal{F} -free Vertex Deletion	NP-c. (Thm. 5.1)	NP-c.*	FPT/PK (Thm. 5.2)	?	?	FPT/? (Thm. 5.7)
Feedback Vertex Set	NP-c. (Thm. 6.1)	NP-c.*	FPT/PK (Thm. 6.2)	?	W[1]-h. (Thm. 6.14)	? / ?
Independent Set		<i>Large Secl.</i> NP-c.*		<i>Large Secl.</i> \rightarrow	\rightarrow	W[1]-h. (Thm. 7.1)

Table 1: Overview of our results. PK stands for polynomial kernel. The results marked by an asterisk follow by a straightforward reduction from the non-secluded variant. The complexity for combinations marked by an arrow are resolved by a stronger result in a farther column.

On the positive side, for SMALL SECLUDED \mathcal{F} -FVD and SMALL SECLUDED s - t -SEPARATOR, we prove fixed-parameter tractability when parameterized by $k + \ell$.

We also study, for two integers $p < q$, the p -secluded version of q -DOMINATING SET (q -DS): a vertex set S is a q -dominating set if every vertex of $V \setminus S$ has distance at most q to some vertex in S . By p -secluded we mean that we upper bound the size of the distance- p -neighborhood of the solution S . This problem admits a complexity dichotomy: Whenever $2p > q$, (SMALL) p -SECLUDED q -DOMINATING SET is fixed-parameter tractable with respect to k (with respect to $k + \ell$), but it is W[2]-hard otherwise.

Finally, we also study the possibility for effective polynomial-time data reduction. We observe that the polynomial-size problem kernels for FEEDBACK VERTEX SET and \mathcal{F} -FREE VERTEX DELETION carry over to their SECLUDED variants, but otherwise we obtain mostly absence of polynomial-size problem kernels unless the polynomial hierarchy collapses.

Related work. SECLUDED PATH and SECLUDED STEINER TREE were introduced and proved NP-complete by Chechik et al. [12]. They obtained approximation algorithms for both problems with approximation factors related to the maximum degree. They also showed that SECLUDED PATH is fixed-parameter tractable with respect to the maximum vertex degree of the input graph, whereas vertex weights lead to NP-hardness for maximum degree four.

Fomin et al. [25] studied the parameterized complexity of SECLUDED PATH and SECLUDED STEINER TREE, showing that both are fixed-parameter tractable even in the vertex-weighted setting. Furthermore, they showed that SECLUDED STEINER TREE is fixed-parameter tractable with respect to $k - s + p$, where p is the number of terminals, k is the desired size of the closed neighborhood of the solution, and s is the size of an optimum Steiner tree. On the other hand this problem is co-W[1]-hard when parameterized by $k - s$ only [25].

The small secluded concept can be found in the context of separator problems in graphs [24, 37]. Marx [37]

introduced the CUTTING k VERTICES problem, which asks, given a graph $G = (V, E)$ and two integers $k \geq 1$ and $\ell \geq 0$, whether there is a non-empty set $S \subseteq V$ such that $|S| \leq k$ and $|N_G(S)| = \ell$. It follows from the work of Bui and Jones [8] that the problem is NP-hard and Marx [37] proved that the problem is W[1]-hard with respect to $k + \ell$. Moreover, for the problem variant where the set S is required to induce a connected subgraph in G , he proved that it becomes fixed-parameter tractable when parameterized by $k + \ell$, while staying W[1]-hard with respect to k or ℓ . Fomin et al. [24] studied the variant of CUTTING k VERTICES where $|N_G(S)| \leq \ell$, thus resembling our small secluded concept. They proved that this variant is W[1]-hard with respect to k , but becomes fixed-parameter tractable when parameterized by ℓ . As to the latter, we remark that for none of our studied small secluded problems we observed fixed-parameter tractability with respect to ℓ . Somewhat surprisingly, if it is additionally required that S has to contain a predefined vertex $s \in V$, then this variant becomes W[1]-hard with respect to ℓ while staying fixed-parameter tractable when parameterized by $k + \ell$ [24].

The concept of *isolation* states that the solution should have few edges to the rest of the graph and was originally introduced for finding cliques [31]. Isolation was subsequently explored also for more general definitions of dense subgraphs [28, 30, 31, 34]. Chiefly the constraint that the vertices in the solution shall have maximum/minimum/average outdegree bounded by a parameter was considered [28, 31, 34], leading to various parameterized tractability and hardness results. Also the overall number of edges outgoing the solution has been studied recently [30]. Finding isolated vertices without constraint on their topology was already studied by Downey et al. [17].

SMALL SECLUDED II and LARGE SECLUDED II can be seen as special cases of FIXED CARDINALITY OPTIMIZATION [7, 10, 11, 33]. Hence, we can derive some corollaries for secluded problems from results from the literature on FIXED CARDINALITY OPTIMIZATION, see below.

Preliminary observations. Concerning the classical computational complexity, the SMALL (LARGE) SECLUDED variant of a problem is at least as hard as the nonsecluded problem, by a simple reduction in which we set $\ell = n$, where n denotes the number of vertices in the graph. Since this reduction is a parameterized reduction with respect to k , parameterized hardness results for this parameter transfer, too. Furthermore, observe that hardness also transfers from SECLUDED II to SMALL SECLUDED II for all predicates II, since SECLUDED II allows for a parameterized Turing reduction to SMALL SECLUDED II: try out all k' and ℓ' with $k = k' + \ell'$.

Observation 1.1. SECLUDED II parameterized by k is parameterized Turing reducible to SMALL SECLUDED II parameterized by $(k + \ell)$ for all predicates II.

Additionally, many tractability results (in particular polynomial time solvability and fixed-parameter tractability) transfer from SMALL SECLUDED II parameterized by $(k + \ell)$ to SECLUDED II parameterized by k . Therefore, for the SMALL (LARGE) SECLUDED variants of the problems the interesting cases are those where the base problem is tractable (deciding whether input graph G contains a vertex set S of size k that satisfies $\Pi(G, S)$) or where the size ℓ of the open neighborhood is a parameter.

SMALL SECLUDED II can be solved using techniques for FIXED CARDINALITY OPTIMIZATION [7, 10], where we are given an objective function ϕ over subsets of some universe (usually via an oracle or efficient algorithm) and we seek to optimize it over all k -element subsets: Define the universe as the vertex set in the input graph, and set $\phi(S) = \infty$ whenever $\Pi(G, S)$ is not satisfied. Otherwise $\phi(S) = |N(S)|$. Iterating over all $i = 1, \dots, k$, minimizing ϕ for i -vertex subgraphs, and checking whether a solution S with $\phi(S) \leq \ell$ exists thus solves SMALL SECLUDED II.

In SMALL SECLUDED II no vertex of degree greater than $k + \ell$ can be part of the solution. Hence, the above formulation makes SMALL SECLUDED II amenable to the random separation framework for FIXED CARDINALITY OPTIMIZATION in graphs with bounded degree [11]: Roughly, we can restrict the domain of ϕ to the set of vertices of degree smaller than $k + \ell$, effectively removing large-degree vertices from the graph, while their contribution to satisfying II and to the neighborhoods of other vertices is still present in ϕ . Applying then a fixed-parameter algorithm with respect to the maximum degree for FIXED CARDINALITY OPTIMIZATION given by Cai et al. [11, Theorem 4], we can thus derive fixed-parameter algorithms with

respect to $k + \ell$ for some SMALL SECLUDED Π problems. Herein, $\Pi(G, S)$ must be computable in fixed-parameter time and fulfill certain conditions which roughly state that subgraphs of G that fulfill Π and have a certain distance from each other in G can be combined via the disjoint union into one subgraph fulfilling Π . For example, finding secluded subgraphs of fixed minimum degree, subgraphs of even degree, subgraphs that induce matchings, and subgraphs of fixed diameter is fixed-parameter tractable with respect to $k + \ell$ through that approach. Furthermore, as a special case we obtain that, if $\Pi(G, S)$ is fixed-parameter tractable with respect to $k + \ell$ and each vertex subset S satisfying $\Pi(G, S)$ is connected, then SMALL SECLUDED Π is fixed-parameter tractable with respect to $k + \ell$.

Note that a similar strategy as above does not work for LARGE SECLUDED Π , because we cannot bound the maximum degree in the same fashion: Neighbors of a solution vertex may be taken into the (arbitrarily large) solution to keep the neighborhood of the solution small.

Organization. We give basic definitions from graph theory and parameterized complexity theory in Section 2. Each of the subsequent sections is dedicated to one of the studied problems. We study s - t -SEPARATOR in Section 3, q -DOMINATING SET in Section 4, \mathcal{F} -FREE VERTEX DELETION in Section 5, FEEDBACK VERTEX SET in Section 6, and INDEPENDENT SET in Section 7. Section 8 summarizes results and gives directions for future research.

2. Preliminaries

We use standard notions from parameterized complexity [14, 18, 22, 40] and graph theory [15, 43]. We use $[p]$ to denote the set $\{1, \dots, p\}$.

Graph theory. Let $G = (V, E)$ be an undirected graph (all graphs in this paper are undirected). We denote by $V(G)$ the vertex set of G and by $E(G)$ the edge set of G . For a vertex set $W \subseteq V(G)$ (edge set $F \subseteq E(G)$), we denote by $G[W]$ ($G[F]$) the subgraph of G induced by the vertex set W (edge set F), respectively. We also denote by $G - W$ the graph $G[V(G) \setminus W]$.

A graph with vertex set $\{v_0, \dots, v_x\}$ and edge set $\{\{v_{i-1}, v_i\} \mid i \in [x]\}$ is called a path (with endpoints v_0 and v_x , also referred to as v_0 - v_x path). The length of a path is the number of edges. A graph with vertex set $\{v_0, \dots, v_x\}$ and edge set $\{\{v_{i-1}, v_i\} \mid i \in [x]\} \cup \{v_x, v_0\}$ is called a cycle (of length $x + 1$).

We denote $d_G(u, v)$ the *distance* between vertices u and v in G , that is, the number of edges of a shortest u - v path in G . For a set V' of vertices and a vertex $v \in V$ we let the distance of v from V' be $d_G(v, V') := \min\{d_G(u, v) \mid u \in V'\}$. We use $N_G^d[V'] = \{v \mid d_G(v, V') \leq d\}$ and $N_G^d(V') = N_G^d[V'] \setminus V'$ for any $d \geq 0$ (hence $N_G^0(V') = \emptyset$). We omit the index if the graph is clear from context and also use $N[V']$ for $N^1[V']$ and $N(V')$ for $N^1(V')$. If $V' = \{v\}$, then we write $N^d[v]$ in place of $N^d[\{v\}]$. The *diameter* of a graph G is the maximum distance between v and w over all $v, w \in V(G)$.

A subset $V' \subseteq V(G)$ is called an s - t separator in G for two distinct vertices s and t if there is no s - t path in $G - V'$. An s - t separator V' is called minimal if for all $V'' \subsetneq V'$ holds that there is an s - t path in $G - V''$.

Parameterized complexity. Parameterized complexity has been introduced to more effectively but optimally solve NP-hard problems: one accepts the apparently inevitable combinatorial explosion in algorithms for NP-hard problems, yet decouples it from the input size and limits it to one aspect of the problem—some small *parameter*. The instances (x, k) of a *parameterized problem* $P \subseteq \Sigma^* \times \mathbb{N}$ consist of an input x and a parameter k . A parameterized problem P is *fixed-parameter tractable (FPT)* if, for every $(x, k) \in \Sigma^* \times \mathbb{N}$, it can be decided in $f(k) \cdot |x|^{O(1)}$ time whether $(x, k) \in P$, where f is an arbitrary computable function only depending on k . The fixed-parameter tractable parameterized problems form the parameterized complexity class FPT.

Parameterized complexity provides means to show intractability: There is a hierarchy of parameterized complexity classes $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{W}[P]$, where all inclusions are conjectured to be strict. A *parameterized reduction* from parameterized problem P_1 to parameterized problem P_2 is an algorithm that maps an instance (x, k) of P_1 to an instance (x', k') of P_2 in time $f(k) \cdot \text{poly}(|x|)$ such that (x, k) and (x', k') are equivalent and $k' \leq g(k)$, where f and g are arbitrary computable functions only depending on k . We say

that an instance (x, k) of parameterized problem P_1 is *equivalent* with an instance (x', k') of parameterized problem P_2 if $(x, k) \in P_1 \iff (x', k') \in P_2$. A *polynomial-parameter transformation* is a parameterized reduction for which both f and g are polynomial. Note that such a reduction is also a polynomial-time many-one reduction from P_1 to P_2 considered in classical complexity theory (also called Karp reduction). A parameterized problem P_2 is *W[t]-hard* if there is a parameterized reduction from each problem $P_1 \in W[t]$ to P_2 . No W[t]-hard problem is fixed-parameter tractable unless $\text{FPT} = W[t]$.

A *parameterized Turing reduction* from parameterized problem P_1 to parameterized problem P_2 is an algorithm that decides whether $(x, k) \in P_1$ in $f(k) \cdot |x|^{O(1)}$ time if it is provided with access to an oracle that can decide instances (x', k') of P_2 with $k' \leq g(k)$ in constant time. Here f and g are arbitrary computable functions only depending on k .

Problem kernelization. Parameterized complexity introduced the concept of kernelization to measure the effect of polynomial-time data reduction. A *kernelization* is a polynomial-parameter transformation of a parameterized problem P to itself, transforming an instance (x, k) of P into an instance (x', k') (the *kernel*), such that $|x'| + k' \leq f(k)$ for some computable function f only depending on k . We call f the *size* of the kernel. If $f \in k^{O(1)}$, then we say that P admits a polynomial-size kernel. A decidable parameterized problem is fixed-parameter tractable if and only if it admits a kernel [14].

Using the following cross-composition method, one can show that a problem does not have polynomial-size kernels unless the polynomial-time hierarchy collapses to the third level.

An equivalence relation \mathcal{R} on the instances of some problem L is a *polynomial equivalence relation* if

- (i) one can decide for any two instances in time polynomial in their sizes whether they belong to the same equivalence class, and
- (ii) for any finite set S of instances, \mathcal{R} partitions the set into at most $(\max_{x \in S} |x|)^{O(1)}$ equivalence classes.

An *OR-cross-composition* of an NP-hard problem L into a parameterized problem P (with respect to a polynomial equivalence relation \mathcal{R} on the instances of L) is an algorithm that takes ℓ \mathcal{R} -equivalent instances x_1, \dots, x_ℓ of L and constructs in time polynomial in $\sum_{i=1}^{\ell} |x_i|$ an instance (x, k) of P such that

- k is polynomially upper-bounded in $\max_{1 \leq i \leq \ell} |x_i| + \log(\ell)$ and
- $(x, k) \in P$ if and only if there is at least one $\ell' \in [\ell]$ such that $x_{\ell'} \in L$.

If an NP-hard problem L OR-cross-composes into a parameterized problem P , then P does not admit a polynomial-size kernel, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [14], which would cause a collapse of the polynomial-time hierarchy to the third level.

3. *s-t*-Separator

In this section, we show that SECLUDED *s-t*-SEPARATOR is in P, while SMALL SECLUDED *s-t*-SEPARATOR is NP-hard and $W[1]$ -hard with respect to the size of the open neighborhood and with respect to the size of the solution. Moreover, we show that, parameterized by the sum of the sizes of the open neighborhood and the solution, the problem is fixed-parameter tractable yet does not allow for polynomial-size kernels.

3.1. Secluded *s-t*-Separator

In this subsection we show that the following problem can be solved in polynomial time.

Problem (SECLUDED *s-t*-SEPARATOR).

Input: A graph $G = (V, E)$, two distinct vertices $s, t \in V$, and an integer k .

Question: Is there an *s-t* separator $S \subseteq V \setminus \{s, t\}$ such that $|N_G[S]| \leq k$?

Theorem 3.1. SECLUDED *s-t*-SEPARATOR can be solved in polynomial time.

In order to prove Theorem 3.1, we show that it is enough to compute a separator of size k in the third power of a graph that is obtained from G by adding two new terminals to s and t :

Definition 3.2. For $x \in \mathbb{N}$ the x -th power of a graph $G = (V, E)$ is a graph $G' = (V, E')$ where for each pair of distinct vertices $u, v \in V$ we have $\{u, v\} \in E'$ if and only if $d_G(u, v) \leq x$.

That is, the x -th power of a graph G is obtained by adding edges between vertices that are at distance at most x in G . This can be done in polynomial time.

Lemma 3.3. Let $G = (V, E)$ be an undirected graph with two distinct vertices $s, t \in V$. Let G'' be the graph obtained from G by adding two vertices s', t' and two edges $\{s', s\}, \{t, t'\}$. Then there is an s - t -separator S in G with $|N[S]| \leq k$ if and only if there is an s' - t' -separator S' with $|S'| \leq k$ in the third power G' of G'' .

Proof. (\Rightarrow) Let S be an s - t -separator in G with $|N[S]| \leq k$. Observe that S then also constitutes an s' - t' -separator in G'' as every path in G'' from s' must go through s and every path to t' must go through t . We claim that $S' = N[S]$ is an s' - t' -separator in G' . Suppose for contradiction that there is an s' - t' path $P = p_0, p_1, \dots, p_q$ in $G' - S'$. Let A' be the set of vertices of the connected component of $G'' - S$ containing s' and let a be the largest index such that $p_a \in A'$ (note that $p_0 = s' \in A'$ and $p_q = t' \notin A'$ by definition). It follows that $p_{a+1} \notin A'$ and, since $\{p_a, p_{a+1}\} \in E'$, there is a p_a - p_{a+1} path P' in G'' of length at most three. As we have $p_a \in A'$ and $p_{a+1} \in V \setminus (A' \cup S')$ and $G[A']$ is a connected component of $G'' - S$, there must be a vertex x of S on P' . Since neither p_a nor p_{a+1} is in $S' = N[S]$, it follows that $d_G(p_a, x) \geq 2$ and $d_G(p_{a+1}, x) \geq 2$. This contradicts P' having length at most 3.

(\Leftarrow) Let S' be an s' - t' -separator in G' of size at most k . Let A' be the vertex set of the connected component of $G' - S'$ containing s' . Consider the set $S = \{v \in S' \mid d_{G''}(v, A') = 2\}$. We claim that S is an s - t -separator in G and, moreover, that $N[S] \subseteq S'$ and, hence, $|N[S]| \leq k$. As to the second part, we have $S \subseteq S'$ by definition. Suppose for contradiction that there was a vertex $u \in N(S) \setminus S'$ that is a neighbor of $v \in S$. Then, since $d_{G''}(v, A') = 2$, we have $d_{G''}(u, A') \leq 3$, meaning that u has a neighbor in A' in G' , and, thus u is in A' . This implies that $d_{G''}(v, A') = 1$, a contradiction. Hence, $|N[S]| \leq k$.

It remains to show that S is an s - t -separator in G . For this, we prove that S is an s' - t' -separator in G'' . Since it contains neither s nor t , it follows that it must be also an s - t -separator in G . Assume for contradiction that there is an s' - t' path in $G'' - S$. This implies that $d_{(G''-S)}(t', A')$ is well defined (and finite). Let $q := d_{(G''-S)}(t', A')$ and let P be a corresponding shortest path in $G'' - S$. Let us denote $P = p_0, \dots, p_q$ with $p_q = t'$ and $p_0 \in A'$. If $d_{G''}(t', A') \leq 3$, then t' has a neighbor in A' in G' , and therefore it is in A' contradicting our assumption that S' is an s' - t' -separator in G' . As $t' = p_q$, we have $q > 3$. Since $d_{G''}(p_0, A') = 0$, $d_{G''}(p_q, A') > 3$, and $d_{G''}(p_{i+1}, A') \leq d_{G''}(p_i, A') + 1$ for every $i \in \{0, \dots, q-1\}$, there is an a such that $d_{G''}(p_a, A') = 2$. Note that each vertex v with $d_{G''}(v, A') \leq 3$ is either in A' or in S' . If p_a is not in S' , then p_a is in A' , contradicting our assumptions on P and q as $a \geq 2$. Therefore we have $d_{G''}(p_a, A') = 2$ and p_a is in S' . It follows that p_a is in S , a contradiction. \square

As a minimum s - t separator can be computed in polynomial time using standard methods, for example, based on network flows (cf. e.g. [32]), Theorem 3.1 follows immediately from Lemma 3.3.

3.2. Small Secluded s - t -Separator

In this subsection we prove hardness and tractability results for the following problem.

Problem (SMALL SECLUDED s - t -SEPARATOR).

Input: A graph $G = (V, E)$, two distinct vertices $s, t \in V$, and two integers k, ℓ .

Question: Is there an s - t separator $S \subseteq V \setminus \{s, t\}$ such that $|S| \leq k$ and $|N_G(S)| \leq \ell$?

We show that, in contrast to SECLUDED s - t -SEPARATOR, the above problem is NP-hard. Moreover, at the same time, we show parameterized hardness with respect to k and with respect to ℓ . Later, however, we will show a fixed-parameter algorithm for the combined parameter $k + \ell$.

Theorem 3.4. SMALL SECLUDED s - t -SEPARATOR is NP-hard and $W[1]$ -hard when parameterized by k or by ℓ .

In the proof of the theorem, we reduce from the CUTTING AT MOST k VERTICES WITH TERMINAL [24] problem, which asks, given a graph $G = (V, E)$, a vertex $s \in V$, and two integers $k \geq 1$, $\ell \geq 0$, whether there is a set $S \subseteq V$ such that $s \in S$, $|S| \leq k$, and $|N_G(S)| \leq \ell$. Fomin et al. [24] proved that CUTTING AT MOST k VERTICES WITH TERMINAL is NP-hard and W[1]-hard when parameterized by k or by ℓ .

Proof. We give a polynomial-parameter transformation from CUTTING AT MOST k VERTICES WITH TERMINAL to SMALL SECLUDED s - t -SEPARATOR.

Construction. Let $\mathcal{I} := (G = (V, E), s, k, \ell)$ be an instance of CUTTING AT MOST k VERTICES WITH TERMINAL. We construct an instance $\mathcal{I}' := (G', s', t', k', \ell')$ of SMALL SECLUDED s - t -SEPARATOR equivalent to \mathcal{I} as follows. To obtain G' from G we add to G two vertices s' and t' and two edges $\{s', s\}$ and $\{s, t'\}$. Note that $G = G' - \{s', t'\}$. We set $k' = k$ and $\ell' = \ell + 2$. Hence, we ask for an s' - t' separator $S \subseteq V(G') \setminus \{s', t'\}$ in G' of size at most k' with $|N_{G'}(S)| \leq \ell'$. Clearly, the construction can be carried out in polynomial time.

Correctness. We show that \mathcal{I} is a yes-instance of CUTTING AT MOST k VERTICES WITH TERMINAL if and only if \mathcal{I}' is a yes-instance of SMALL SECLUDED s - t -SEPARATOR.

(\Rightarrow) Let \mathcal{I} be a yes-instance and let $S \subseteq V(G)$ be a solution to \mathcal{I} , that is, $s \in S$, $|S| \leq k$, and $|N_G(S)| \leq \ell$. We claim that S is also a solution to \mathcal{I}' . Since $s \in S$ and s' and t' are both only adjacent to s , S separates s' from t' in G' . Moreover, $|S| \leq k = k'$ and, as $N_{G'}(S) = N_G(S) \cup \{s', t'\}$, we have $|N_{G'}(S)| \leq \ell + 2 = \ell'$. Hence, S is a solution to \mathcal{I}' , and \mathcal{I}' is a yes-instance.

(\Leftarrow) Let \mathcal{I}' be a yes-instance and let $S' \subseteq V(G') \setminus \{s', t'\}$ be an s' - t' separator in G' with $|S'| \leq k'$ and $|N_{G'}(S')| \leq \ell'$. We claim that S' is also a solution to \mathcal{I} . Note that $|S'| \leq k' = k$. Since S' is an s' - t' separator in G' and s' and t' are both adjacent to s , it follows that $s \in S'$ and $s', t' \in N_{G'}(S')$. Thus, we have $s \in S'$ and $|N_G(S')| = |N_{G' - \{s', t'\}}(S')| = |N_{G'}(S')| - 2 \leq \ell' - 2 = \ell$. Hence, S' is a solution to \mathcal{I} and \mathcal{I} is a yes-instance.

Note that, in the reduction, k' and ℓ' only depend on k and ℓ , respectively. Since CUTTING AT MOST k VERTICES WITH TERMINAL parameterized by k or by ℓ is W[1]-hard [24], it follows that SMALL SECLUDED s - t -SEPARATOR parameterized by k or by ℓ is W[1]-hard. \square

Note that the above reduction exploits the fact that we can increase the separator size to decrease the number of vertices in its neighborhood. In fact, most of the vertices declared to be in the separator do not serve to separate the terminals at all. To avoid this idiosyncrasy we suggest to study secluded inclusion-wise minimal separators in future work.

In the following, we prove that SMALL SECLUDED s - t -SEPARATOR is FPT when parameterized by $k + \ell$.

Theorem 3.5. There is a computable function f such that SMALL SECLUDED s - t -SEPARATOR is solvable in $f(k + \ell) \cdot m \log n$ time, where n is the number of vertices and m the number of edges in the input graph, k is the separator size and ℓ its open neighborhood size.

To prove Theorem 3.5, we exploit that SMALL SECLUDED s - t -SEPARATOR is efficiently solvable on graphs of small treewidth:

Lemma 3.6. There is a computable function f such that SMALL SECLUDED s - t -SEPARATOR can be solved in $f(k, \ell, w) \cdot n$ time on graphs of treewidth w .

Proof. By Courcelle's theorem [13], it is enough to express the existence of an s - t -separator of cardinality k and open neighborhood size ℓ in a formula of the monadic second-order logic of graphs such that this formula has a size depending only on k and ℓ .⁷ More specifically, using Courcelle's theorem, we want to verify whether our graph $G = (V, E)$ of treewidth w satisfies

$$\exists S \subseteq V : \text{separates}(s, S, t) \wedge \text{card-le}(S, k) \wedge \forall N \subseteq V : (\text{open-nh}(N, S) \Rightarrow \text{card-le}(N, \ell))$$

⁷We refrained from giving a tedious dynamic program on a tree-decomposition and opted for Courcelle's theorem instead for clarity and because a singly exponential running time is out of reach due to the use of the treewidth reduction technique that we will apply later.

where the predicate

$$\text{card-le}(S, k) \equiv \forall v_1 \forall v_2 \dots \forall v_{k+1} : \left(\bigwedge_{i=1}^{k+1} v_i \in S \right) \Rightarrow \left(\bigvee_{i=1}^k \bigvee_{j=i+1}^{k+1} v_i = v_j \right)$$

is true if and only if $|S| \leq k$ and this predicate has a size depending only on k ,

$$\text{open-nh}(N, S) \equiv \forall v \in V : v \in N \Leftrightarrow (v \notin S \wedge \exists u \in S : \text{adj}(u, v))$$

is a constant-size predicate that is true if and only if $N = N(S)$, and, finally,

$$\begin{aligned} \text{separates}(s, S, t) \equiv \exists A \subseteq V \exists B \subseteq V : (\forall v : v \in A \vee v \in B \vee v \in S) \wedge s \in A \wedge t \in B \\ \wedge (\forall u \forall v : \text{adj}(u, v) \Rightarrow (u \notin A \vee v \notin B) \wedge (u \notin B \vee v \notin A)) \end{aligned}$$

is a constant-size predicate that is true if and only if S separates s from t . Namely it is true if and only if the vertex set $V \setminus S$ can be divided into two sets A and B such that $s \in A$, $t \in B$, and there are no edges between the sets A and B . \square

In view of Lemma 3.6, to prove Theorem 3.5 it is enough to reduce SMALL SECLUDED s - t -SEPARATOR to the case where the input graph has treewidth bounded by some function in $k + \ell$. To this end, we use the following treewidth reduction technique. Let G be a graph and $W \subseteq V(G)$. The *torso* $\text{torso}(G, W)$ is a graph obtained from $G[W]$ by taking each connected component C in $G \setminus W$ and making $N(C)$ into a clique in $G[W]$. The following lemma is implied by Marx et al.'s Lemma 2.11 [38]:

Lemma 3.7 (Marx et al. [38]). Let s, t be two vertices of a graph G and let $r \in \mathbb{N}$. Let W' be the union of all inclusion-wise minimal s - t separators of size at most r . Then, there is an $f(r) \cdot (n + m)$ -time algorithm that returns a set $W \supseteq W' \cup \{s, t\}$ such that $\text{torso}(G, W)$ has treewidth upper bounded by some function depending only on r .

The algorithm for Theorem 3.5 now proceeds in three stages. First, we use random separation [11]: we randomly color all vertices red or green. With sufficiently high probability, all separator vertices will be colored green and all their neighbors red. We will henceforth assume to have such a coloring. Second, we prepare a graph of low treewidth containing our solution. This we do by first contracting each green component in the graph to make our desired separator an inclusion-wise minimal one. Relying on this, we then use the treewidth reduction technique (Lemma 3.7) to compute a vertex subset containing the solution which induces a graph of small treewidth. In this vertex subset, however, neighbors of some s - t separators may be missing, possibly introducing false positives. Furthermore, paths between s and t may be missing, introducing false positives as well. Hence, we make some modifications to reintroduce the corresponding information. To the resulting graph, we then apply Lemma 3.6.

Proof of Theorem 3.5. In the following we simultaneously describe the algorithm and gather arguments to show that, if there is a solution, then the algorithm will accept. For this purpose, assume that the input is a yes-instance and, among all separators with at most k vertices and open neighborhood of size at most ℓ , let K be a separator with minimum number of vertices. We will argue in the end that, if the input is a no-instance, then the algorithm will reject.

Random separation stage: We present this stage already in derandomized form. We use Naor et al.'s construction of universal sets [39]. An (n, d) -universal set \mathcal{F} over some universe U of size n is a family of subsets of U such that, for each $A \subseteq U$ of size exactly d , the family $\{A \cap S \mid S \in \mathcal{F}\}$ contains A and all subsets of A . Naor et al. [39] showed that an (n, d) -universal set of size $2^d d^{O(\log d)} \log n$ can be computed in $2^d d^{O(\log d)} n \log n$ time. In the random separation stage, we compute an $(n - 2, k + \ell)$ -universal set \mathcal{F} over the vertex set $V \setminus \{s, t\}$ of the input graph. We iterate over all the sets $F \in \mathcal{F}$ and in each such iteration perform all the algorithm steps described later. Call the vertices in F *green* and the vertices in $V \setminus F$ *red*. Note that, in one of these iterations, we have that each vertex in K is green and each vertex in $N(K)$ is red,

by the definition of universal sets. Call such an iteration *good* for K . Our aim is to show that, if we are in a good iteration for K , we will accept.

Building a graph of low treewidth: We call an inclusion-wise maximal connected set of green vertices of G a *green component* of G . We construct a graph G_1 from G by contracting each green component into one vertex. We remove all self-loops introduced in the process. In the following, we call *component vertices* the vertices in G_1 corresponding to green components in G . Let $m : V(G) \rightarrow V(G_1)$ be the function mapping each red vertex to itself and each green vertex to the corresponding component vertex.

If we are in a good iteration for K , each green component in G is either contained in K or disjoint from it. We claim that then K induces an inclusion-wise minimal s - t separator K' in G_1 , where $K' = m(K)$, that is, we obtain K' from K by replacing each green component in K by the corresponding component vertex in G_1 . Clearly, K' is an s - t separator in G_1 . Observe that K' is an independent set in G_1 . Now, for the sake of a contradiction, assume that there is an inclusion-wise minimal separator K'_1 strictly contained in K' . Since K' is an independent set, $|N_{G_1}(K'_1)| \leq |N_{G_1}(K')| \leq \ell$. Hence, $m^{-1}(K'_1)$ is an s - t separator in G with $|m^{-1}(K'_1)| < |K| \leq k$ and $|N_G(m^{-1}(K'_1))| = |N_{G_1}(K'_1)| \leq \ell$. This is a contradiction to the fact that K has minimum number of vertices among s - t separators in G with at most k vertices and neighborhood size at most ℓ . Thus, indeed, K' is inclusion-wise minimal.

We next use treewidth reduction (Lemma 3.7) to compute a vertex set W containing the vertices s, t and all inclusion-wise minimal s - t separators in G_1 of size at most k such that $\text{torso}(G_1, W)$ has treewidth upper bounded by a function of k alone. Clearly, $K' \subseteq W$. In the following, we construct a graph G_4 from $G_1[W]$ so that we can simply compute a small secluded s - t separator in G_4 instead of the input graph. (We bypass $\text{torso}(G_1, W)$ and use it only to show that G_4 has bounded treewidth.) The construction of G_4 is as follows.

- 1) Initially, let $G_2 = G_1[W]$.
- 2) Denote $B = N_G(m^{-1}(W))$. Add B to G_2 and for each $b \in B$ and each $v \in N_G(b)$ if $m(v)$ is in W , then make b adjacent to $m(v)$.
- 3) Let \mathcal{C} be the set of connected components of the graph $G_1 - W$. For each $C \in \mathcal{C}$ introduce the new vertex c_C to G_2 and make it adjacent to each vertex in $m^{-1}(C) \cap B$. For reference later on, let $D = \{c_C \mid C \in \mathcal{C}\}$.
- 4) Let us now distinguish the following subsets of W . Recall that F is the set of green vertices and $m(F)$ is the set of component vertices. The set $R = W \setminus m(F)$ is the set of red vertices in W . The set $H = \{a \in W \mid |m^{-1}(a)| > k\}$ is the set of component vertices in W corresponding to huge green components that cannot take part in the solution. Finally, the set $A = \{a \in m(F) \cap W \mid |m^{-1}(a)| \leq k\} = W \setminus (R \cup H)$ is the set of component vertices in W allowed in the solution.

Construct G_3 from G_2 as follows. Replace each component vertex a in A by a clique on vertex set $m^{-1}(a)$ and make each vertex of the clique adjacent to each neighbor of a in G_2 . Note that, as the neighbors of a green component are all red, none of the neighbors are being replaced in the current step. Let $A' = m^{-1}(A) \subseteq F$ be the set of vertices newly (re-)introduced to G_3 .

- 5) Finally, to obtain G_4 from G_3 , for each vertex in $R \cup H \cup B \cup D$ introduce $k + \ell + 1$ new degree-one neighbors.

We claim that G_4 satisfies the following properties.

- (i) If we are in a good iteration for K , then $K \subseteq V(G_4)$, $|N_{G_4}(K)| \leq \ell$, and K is an s - t separator in G_4 .
- (ii) If there is an s - t separator L in G_4 of size at most k and $|N_{G_4}(L)| \leq \ell$, then there is also such a separator in G . (Regardless of whether we are in a good iteration.)
- (iii) There is a function f such that the treewidth of G_4 is bounded from above by $f(k)$.

Let us prove Property (i). Assume that we are in a good iteration for K . Recall that $K' \subseteq W$. Hence, K' is present in G_2 after Step 1 and, clearly, is not modified in Steps 2 and 3. Furthermore, each component vertex in K' corresponds to a green connected component in G of size at most k . Hence, in

Step 4, K' is replaced by K and thus $K \subseteq V(G_4)$. Furthermore, $N_{G_4}(K) \subseteq W \cup B$. By construction we have $N_{G_4}(K) \cap W = N_{G_3}(K) \cap W = N_{G_2}(K') \cap W = N_{G_1}(K') \cap W$. Thus, for each vertex $r \in N_{G_4}(K) \cap W$ there is a vertex $v \in K$ such that r and v are adjacent in G by the construction of G_1 . Also, by construction, we have $N_{G_4}(K) \cap B = N_{G_3}(K) \cap B = N_{G_2}(K') \cap B$ and for each $b \in N_{G_4}(K) \cap B$ there is a vertex $v \in K$ such that b and v are adjacent in G by the construction of G_2 . Hence, $N_{G_4}(K) \subseteq N_G(K)$ and $|N_{G_4}(K)| \leq \ell$.

To prove Property (i) it remains to show that K is an s - t separator in G_4 . Suppose for contradiction that there is an s - t path P in $G_4 - K$. Since G_4 and G_3 only differ in degree-one vertices, P is an s - t path also in G_3 . If P uses vertices in $A' = \{m^{-1}(a) \mid a \in A\}$, then we can replace a part of P between the first and last vertex in $m^{-1}(a)$ by a . This way we obtain an s - t path P_2 in $G_2 - K'$. If P_2 uses vertices outside W then let Q be the set of vertices appearing on P_2 between two consecutive vertices a_1 and a_2 of W and observe that $Q \subseteq B \cup D$. We claim that we can remove Q from P_2 and replace it by a path P' between a_1 and a_2 in G_1 and, hence, obtain a path P_1 in G_1 . To see this, note that the vertices $b \in B$ are only connected to vertices in $W \cap N_{G_1}(C)$ and to c_C for a connected component C of $G_1 - W$ with $m(b) \in C$. Furthermore, $c_C \in D$ is only connected to vertices in B . It follows that there is a connected component C of $G_1 - W$ such that $Q \subseteq \{c_C\} \cup \{b \in B \mid m(b) \in C\}$ and $\{a_1, a_2\} \subseteq N_{G_1}(C)$. Hence, there is the claimed path P' between a_1, a_2 in G_1 and we can replace Q in P_2 with P' . Doing this with each part of P_2 outside W , we obtain an s - t path in $G_1 - K'$, a contradiction. Hence, K is an s - t separator in G_4 .

We now prove Property (ii). We may assume that L does not contain any vertex that has been introduced in Step 5 as they could be removed from L , yielding another separator that fits the definition of L . Furthermore, L does not contain any vertex in $R \cup H \cup B \cup D$, because L 's neighborhood has size at most ℓ . Hence, we have $L \subseteq A' \subseteq V(G)$.

Suppose for contradiction that there is an s - t path P in $G - L$. As shown above, no (part of a) green component of size more than k in G is contained in L . Furthermore, since green components of size at most k have been contracted in G_1 , and then replaced by a clique of vertices with identical neighborhood in G_3 , we may assume without loss of generality, that each green component of G is either completely contained in L or disjoint from it. Hence, contracting green parts of path P we obtain an s - t path P_1 in $G_1 - m(L)$. For each part of P_1 using vertices outside of W we do the following. Obviously, each such part must stay within one connected component C of $G_1 - W$. Let b_1 and b_2 be the first and last vertex of C on P_1 . We replace the part of P_1 between b_1 and b_2 by the vertex c_C . Doing this with each part of P_1 outside of W , we obtain an s - t path P_2 in $G_2 - m(L)$. If we replace each vertex $a \in A$ on P_2 with an arbitrary vertex of $m^{-1}(a)$, we obtain an s - t path P_3 in $G_3 - L$. This path is also present in $G_4 - L$, contradicting L being an s - t separator in G_4 . Hence, L is an s - t separator in G .

To prove Property (ii) it remains to show that the neighborhood of L in G has size at most ℓ . We show that $N_G(L) \subseteq N_{G_4}(L)$. Recall that $L \subseteq A'$. Let b be a vertex of $N_G(L)$ and v one of its neighbors in L . Obviously $m(v)$ is in $A \subseteq W$ and $|m^{-1}(m(v))| \leq k$. If b is in F , then $m(b) = m(v)$ and b is in $N_{G_4}(L)$, since $m^{-1}(m(v))$ is a clique in G_4 . Otherwise, $m(b) = b$. If b is in W , then b is a neighbor of $m(v)$ in $G_1[W]$, yielding that b is in $N_{G_4}(L)$. Finally, if b is in $G_1 - W$, then b is in B , b is adjacent to $m(v)$ in G_2 , meaning that b is in $N_{G_4}(L)$. Hence, $N_G(L) \subseteq N_{G_4}(L)$, implying that $|N_G(L)| \leq \ell$.

We now prove Property (iii). By Lemma 3.7 there exists a tree decomposition T for $\text{torso}(G_1, W)$ and a function f' such that T has width $f'(k)$. We show how to adapt T into a tree decomposition for G_4 without increasing its width too much. Clearly, T is also a tree decomposition for $G_1[W]$. Recall that the neighborhood of each connected component C of $G_1 - W$ is a clique in $\text{torso}(G_1, W)$. Hence, this neighborhood $N_{G_1}(C)$ occurs in one bag Q of T . Thus, to incorporate c_C and its adjacent edges into T , we make a copy Q' of bag Q , add c_C to Q' and make Q' a child of Q in T . Then for each $b \in B$ adjacent to c_C in G_2 we add a child bag of Q' with vertex set $Q' \cup \{b\}$. Since all neighbors of b in G_2 are in Q' and vertices of D only have neighbors in B , we obtain a valid tree decomposition for G_2 . This increases the width of T by at most 2. In Step 4 we can replace each component vertex by the at most k vertices in the corresponding green component in G , increasing the width of T by at most a factor of k and obtaining a tree decomposition for G_3 . Then for each vertex v to receive degree-one neighbors in Step 5 we find an arbitrary bag Q of T containing v and for each of the degree one neighbors of v we create a child bag of Q containing v and the degree one vertex. Since each of these bags is of size 2, this does not increase the width of T . Making these changes to T , we obtain a tree decomposition for G_4 of width at most $f'(k) \cdot k + 2$, as required.

Conclusion: Let us show that the above properties together with Lemma 3.6 conclude the proof. In each iteration of the random separation phase, we build the graph G_4 and the tree decomposition for it and run the algorithm of Lemma 3.6. If any of the iterations accepts, then we accept, otherwise we reject. On one hand, if K is an s - t separator in G of size at most k and $|N_G(K)| \leq \ell$, then at least one iteration is good for it and the algorithm will accept the input by Property (i). On the other hand, if the algorithm accepts, then there is an s - t separator L in G_4 of size at most k and $|N_{G_4}(L)| \leq \ell$ and by Property (ii) L is such a separator in G . Thus, the algorithm accepts if and only if we face a yes-instance.

The running time can be upper bounded as follows. Computing the universal set \mathcal{F} takes $2^{k+\ell}(k+\ell)^{O(\log(k+\ell))}n \log n$ time. For each of the $2^{k+\ell}(k+\ell)^{\log(k+\ell)} \log n$ elements we make one iteration, each of which takes the following computation time.

Graph G_1 can be constructed in linear time, by first finding the green components in linear time. Then, we compute the neighborhood for all green components simultaneously by scanning over the adjacency lists of each contained vertex, and marking it as neighbor for that component. Finally, we scan over all vertices, finding the neighborhoods of the corresponding component vertices.

Next, Step 1 can clearly be computed in linear time and simultaneously we can find in the same fashion as before the connected components in $G_1 - W$, their neighborhoods in W , as well as the set B together with, for each vertex in B its connected component in $G_1 - W$. Using this information, we can compute Step 2 and 3 in linear time.

Since each vertex in A corresponds to a connected component of size at most k , Step 4 can be computed in $O(k(n+m))$ time. Step 5 can be computed in $O((k+\ell)(n+m))$ time, because each of the sets R, H, B, D can be computed in linear time alongside the previous computations.

Hence, for each element of the universal set we have made a constant number of $O((k+\ell)(n+m))$ -time computations so far. After that, we find a tree decomposition for G_4 in $f(k) \cdot (n+m)$ time, using a fixed-parameter constant-factor approximation algorithm [6] (recall that G_4 has treewidth bounded by a function of k by Property (iii)). Finally, Lemma 3.6 again uses fixed-parameter linear time with respect to $k+\ell$. Thus, each of the steps has $f(k+\ell) \cdot (n+m)$ running time. \square

In contrast to Theorem 3.5, we show that, under standard assumptions, the problem does not admit a polynomial-size kernel with respect to the parameter $k+\ell$:

Theorem 3.8. Unless $\text{NP} \subseteq \text{coNP/poly}$, SMALL SECLUDED s - t -SEPARATOR parameterized by $k+\ell$ does not admit a polynomial kernel.

Proof. We apply an OR-cross-composition with input problem SMALL SECLUDED s - t -SEPARATOR (SSstS) to SMALL SECLUDED s - t -SEPARATOR parameterized by $k+\ell$.

Let $(\mathcal{I}_q = (G_q, s_q, t_q, k_q, \ell_q))_{q=1, \dots, p}$ be instances of SMALL SECLUDED s - t -SEPARATOR. We assume that $\min\{k_i, \ell_i\} \geq 0$ and $\max\{k_i, \ell_i\} \leq |V(G_i)|$ for each $i \in [p]$, because otherwise, we can decide \mathcal{I}_i in polynomial time. By virtue of choosing a corresponding polynomial equivalence relation, we also assume that (i) $0 \leq k_i, \ell_i \leq |V(G_i)|$ for all $i \in [p]$, and (ii) $k_i = k_j$ and $\ell_i = \ell_j$ for all $i, j \in [p]$. We OR-cross-compose into one instance $\mathcal{I} = (G, s_1, t_p, k, \ell)$ of SMALL SECLUDED s - t -SEPARATOR, with $k := k_i$ and $\ell := \ell_i$ for any $i \in [p]$.

Construction: Initially, let G be the disjoint union of G_1, \dots, G_p , that is $G = G_1 \cup \dots \cup G_p$. Identify each t_q with s_{q+1} for all $q \in [p-1]$. Call the obtained vertex st_q , $q \in [p-1]$. For each st_q , $q \in [p-1]$, add $k+\ell+1$ vertices $x_1^q, \dots, x_{k+\ell+1}^q$, and connect them by an edge with st_q . We will also refer to s_1 as s and as st_0 and to t_p as t and as st_p . This finishes the construction of the instance.

Correctness: We claim that \mathcal{I} is a yes-instance if and only if there exists $q \in [p]$ such that \mathcal{I}_q is a yes-instance.

(\Leftarrow) Let $q \in [p]$ such that \mathcal{I}_q is a yes-instance of SSstS. Let $S \subseteq V(G_q) \setminus \{s_q, t_q\}$ be an s_q - t_q -separator of size at most k in G_q such that $N_{G_q}(S) \leq \ell$. By construction of G , for all $V' \subseteq V(G_r) \setminus \{s_r, t_r\}$, $r \in [p]$, it holds that $N_G(V') = N_{G_r}(V')$. Moreover, since G is obtained by a “serial” composition of $\mathcal{I}_1, \dots, \mathcal{I}_p$, every s - t path in G contains $s = st_0, st_1, \dots, st_{p-1}, st_p = t$ in this order. Hence, any vertex set $V' \subseteq V(G_r) \setminus \{s_r, t_r\}$ separating st_{r-1} and st_r in G , $r \in [p]$, also separates s and t in G . Altogether, S is an s - t -separator in G of size at most k with $N_G(S) = N_{G_q}(S) \leq \ell$. Thus, \mathcal{I} is a yes-instance of SSstS.

(\Rightarrow) Let $S \subseteq V(G) \setminus \{s, t\}$ be a minimal s - t separator (of size at most k) such that $N_G(S) \leq \ell$. Observe that $S \cap \{st_1, \dots, st_{p-1}\} = \emptyset$, since every st_r , $r \in [p-1]$, is incident to at least $k + \ell + 1$ vertices. Moreover, no vertex x_j^i , $i \in [p-1]$, $j \in [k + \ell + 1]$ is contained in S since S is chosen as minimal and x_j^i is of degree one and hence not participating in any minimal s - t separator in G . We claim that there exists a $q \in [p]$ with $S \subseteq V(G_q) \setminus \{s_q, t_q\}$. Following the argumentation above, since S separates s and t , there is at least one $r \in [p]$ such that S separates st_{r-1} and st_r . Let q be the minimal index such that S separates st_{q-1} and st_q . Suppose there is an $r \neq q$ such that $S \cap V(G_r) \setminus \{s_r, t_r\} \neq \emptyset$. Since S separates s from st_q , $S' = S \cap (V(G_q) \setminus \{s_q, t_q\})$ is an s - t separator of G of size smaller than S . This contradicts the minimality of S . Hence, $S \subseteq V(G_q) \setminus \{s_q, t_q\}$. Since S separates st_{q-1} and st_q in G , it follows that S separates s_q and t_q in G_q . Together with $|S| \leq k$ and $N_{G_q}(S) = N_G(S)$ implying $|N_G(S)| \leq \ell$, it follows that \mathcal{I}_q is a yes-instance. \square

4. q -Dominating Set

In this section, for two constants $p, q \in \mathbb{N}$ with $0 \leq p < q$, we study the following problems:

Problem (p -SECLUDED q -DOMINATING SET).

Input: A graph $G = (V, E)$ and an integer k .

Question: Is there a set $S \subseteq V$ such that $V = N_G^q[S]$ and $|N_G^p[S]| \leq k$?

Problem (SMALL p -SECLUDED q -DOMINATING SET).

Input: A graph $G = (V, E)$ and two integers k, ℓ .

Question: Is there a set $S \subseteq V$ such that $V = N_G^q[S]$, $|S| \leq k$, and $|N_G^p(S)| \leq \ell$?

For $p = 0$, the size restrictions in both cases boil down to $|S| \leq k$. This is the well-known case of q -DOMINATING SET (also known as q -CENTER) which is NP-hard and W[2]-hard with respect to k (see Loksh-tanov et al. [36], for example). Therefore, for the rest of the section we focus on the case $p > 0$. Additionally, by a simple reduction from q -DOMINATING SET, letting $\ell = |V(G)|$, we arrive at the following observation.

Observation 4.1. For any $0 < p < q$, SMALL p -SECLUDED q -DOMINATING SET is W[2]-hard with respect to k .

Furthermore, let us make the following observation which we will use at multiple occasions in the proofs.

Observation 4.2. Let $(G = (V, E), k)$ be a yes-instance of p -SECLUDED q -DOMINATING SET and $S \subseteq V$ be a q -dominating set of G with $|N_G^p[S]| \leq k$. If G contains a clique C with $|C| > k$, then for any $v \in V$ and any $c \in C$ such that $d_G(v, c) < p$ we have $v \notin S$.

Proof. Let S be a p -secluded q -dominating set of G and C be a clique in G with $|C| > k$. Assume for contradiction that there is a vertex $v \in S$ and a vertex $c \in C$ with $d_G(v, c) < p$. Then we have that $d_G(v, c') \leq p$ for any $c' \in C$, which implies that $C \subseteq N_G^p[S]$ and hence $|N_G^p[S]| > k$, a contradiction. \square

We now go on to show NP-hardness and W[2]-hardness with respect to k for p -SECLUDED q -DOMINATING SET. We reduce from the following problem:

Problem (SET COVER).

Input: A finite universe U , a family $F \subseteq 2^U$, and an integer k .

Question: Is there a subset $X \subseteq F$ such that $|X| \leq k$ and $\bigcup_{x \in X} x = U$?

We write $\bigcup X$ short for $\bigcup_{x \in X} x$. It is known that SET COVER is NP-complete, W[2]-hard with respect to k , and admits no polynomial kernel with respect to $|F|$, unless $\text{NP} \subseteq \text{coNP}/\text{poly}$ [16].

Theorem 4.3. For any $0 < p < q$, p -SECLUDED q -DOMINATING SET is NP-hard. Moreover, it does not admit a polynomial kernel with respect to k , unless $\text{NP} \subseteq \text{coNP}/\text{poly}$.

Proof. We give a polynomial-parameter transformation from SET COVER parameterized by $|F|$. Let (U, F, k) be an instance of SET COVER. Without loss of generality we assume that $0 \leq k < |F|$.

Construction. Let $k' = p + 1 + |F| \cdot p + k$. We construct the graph G of a p -SECLUDED q -DOMINATING SET instance (G, k') as follows. We start the construction by taking two vertices s and r and three vertex sets $V_U = \{u \mid u \in U\}$, $V_F = \{v_A \mid A \in F\}$, and $V'_F = \{v'_A \mid A \in F\}$. We connect vertex r with vertex s by a path of length exactly q . For each $A \in F$ we connect vertices v_A and r by an edge and vertices v_A and v'_A by a path $t_0^A, t_1^A, \dots, t_p^A$ of length exactly p , where $t_0^A = v_A$ and $t_p^A = v'_A$. Let us denote T the set of vertices on all these paths (excluding the endpoints). All introduced paths are internally disjoint and the internal vertices are all new. We connect a vertex $v'_A \in V'_F$ with a vertex $u \in V_U$ by an edge if and only if $u \in A$. Furthermore, we introduce a clique C_U of size k' and make all its vertices adjacent to each vertex in $V'_F \cup V_U$.

If $q - p \geq 2$, then for each $u \in U$ we create a path $b_0^u, b_1^u, \dots, b_{q-p-2}^u$ of length exactly $q - p - 2$ such that $b_0^u = u$ and the other vertices are new. Let us denote the set of all new vertices introduced in this step B . Furthermore, in this case, for each $h \in \{0, \dots, q - p - 2\}$ we introduce a clique C_h^u of size k' and make all its vertices adjacent to vertex b_h^u . Let us denote the set of all vertices introduced in this step C . If $q - p = 1$ we do not introduce any new vertices.

Correctness: We show that the original instance of SET COVER is a yes-instance if and only if the constructed instance of p -SECLUDED q -DOMINATING SET is a yes-instance.

(\Rightarrow) Let us next show that if there is an $X \subseteq F$ such that $|X| \leq k$ and $\bigcup X = U$, then there is a subset S of $V(G)$ such that $V(G) = N_G^q[S]$, and $|N_G^p[S]| \leq k'$. Indeed, take a set $S = \{r\} \cup \{v_A \mid A \in X\}$. It is a routine to check that all the following vertices are in distance at most q to r and, hence, they are in $N_G^q[S]$: vertex s and the vertices on the path between r and s , vertices in V_F and V'_F and vertices on the paths between them. For each $u \in V_U$ there is $A \in X$ such that $u \in A$ (as $\bigcup X = U$). As u is adjacent to v'_A , u is in distance (at most) $p + 1 \leq q$ to v_A and, thus, it is in $N_G^q[S]$. The same holds for vertices of C_U . Moreover, for $q - p \geq 2$, each vertex in $B \cup C$ is in distance at most $q - p - 1$ to some vertex in V_U . Therefore, all vertices in $B \cup C$ are in $N_G^q[S]$ and $V(G) = N_G^q[S]$.

It remains to bound $|N_G^p[S]|$. The p -neighborhood of r is formed by p vertices on the path to s , the vertices in V_F , and $p - 1$ vertices on each path between V_F and V'_F , hence the closed p -neighborhood is of size exactly $p + 1 + |F| \cdot p$. For each vertex in V_F , its p -neighborhood is only formed by vertices already in the neighborhood of r , except for the corresponding vertex in V'_F . We get that $|N_G^p[S]| \leq p + 1 + |F| \cdot p + k$.

(\Leftarrow) For the other direction, let us assume that there is a subset S of $V(G)$ such that $V(G) = N_G^q[S]$ and $|N_G^p[S]| \leq k'$.

We first observe that $S \cap (B \cup C \cup T \cup V'_F \cup V_U \cup C_U) = \emptyset$. Notice that for every $b_h^u \in B$ we have that $\{b_h^u\} \cup C_h^u$ is a clique of size $k' + 1$ and hence, by Observation 4.2, for $v \in B \cup C$ we have that $v \notin S$. Similarly, for any $v'_A \in V'_F$ we have that $\{v'_A\} \cup C_U$ is a clique of size $k' + 1$ and furthermore we have that for each $t \in T$ there is a $v'_A \in V'_F$ such that $d_G(t, v'_A) < p$. It follows by Observation 4.2 that for every $v \in T \cup V'_F \cup C_U$ we have that $v \notin S$. For any $u \in V_U$ we have that $\{u\} \cup C_U$ is a clique of size $k' + 1$. Observation 4.2 yields that $S \cap V_U = \emptyset$. Hence, S may only contain vertices in V_F and the vertices on the path between s and r (including endpoints).

Now suppose $r \notin S$. Then S contains one vertex of V_F for each vertex of V'_F (since these are the only vertices to dominate V'_F and each vertex of V_F can dominate one vertex of V'_F) and at least one vertex of the path between s and r (as no vertex in V_F dominates s). It follows that $|N_G^p[S]| \geq p + 1 + |F| \cdot (p + 1) > k'$, a contradiction. Hence S must contain r . Note that for $S = \{r\}$ we have that $|N_G^p[S]| = p + 1 + |F| \cdot p = k' - k$. Adding a vertex from V_F to S increases the size of the closed p -neighborhood of S by one. It follows that S may contain at most k vertices from V_F .

Now, let us denote $X = \{A \mid v_A \in S\}$. We claim that X is a set cover for U of size at most k , that is, $|X| \leq k$ and $\bigcup X = U$. We already know that $|X| \leq k$. Suppose X is not a set cover. Then there is a $u \in U \setminus \bigcup X$. If $q - p = 1$, then, since S is a solution, there must be a vertex in S in distance at most q from u in G . As $V_U \cap S = \emptyset$, this vertex must be in distance at most $q - 1 = p$ from v'_A for some $A \in F$, $u \in A$. If $q - p \geq 2$, then, since S is a solution, there must be a vertex in S in distance at most q from any vertex in C_{q-p-2}^u in G . As $(T \cup V_U \cup C_U \cup B \cup C) \cap S = \emptyset$, this vertex must also be in distance at most $q - (q - p - 1) - 1 = p$ from v'_A for some $A \in F$, $u \in A$. In both cases it follows that the vertex of S is

path of length exactly q and c with r by a path of length exactly p . We connect a vertex $v'_A \in V'_F$ with a vertex $u \in V_U$ by an edge if and only if $u \in A$.

We introduce a clique C_r of size k' and make all its vertices adjacent to r , and we introduce a clique C_U of size k' and make all its vertices adjacent to each vertex in $V'_F \cup V_U$. If $q - p \geq 2$, then for each $u \in V_U$ we further introduce a path $b_0^u, b_1^u, \dots, b_{q-p-2}^u$, where $b_0^u = u$ and b_h^u is a new vertex for each $h \geq 1$. Let B denote the set of all these vertices. Furthermore, in such a case, let us introduce for each $u \in V_U$ a set of cliques $C_0^u, C_1^u, \dots, C_{q-p-2}^u$, each of size k' , and for each $h \in \{0, \dots, q - p - 2\}$ connect every vertex in C_h^u by an edge to b_h^u . Let C denote the union of vertices in these cliques.

Then we connect for each $A \in F$ the vertex $v_A \in V_F$ with the vertex $v'_A \in V'_F$ by a path $t_0^A, t_1^A, \dots, t_p^A$ of length p , where $t_0^A = v_A$ and $t_p^A = v'_A$. Let T denote the set of vertices on all these paths (excluding the endpoints).

Now let us distinguish two cases. If $3p \geq q$, then we connect r by paths of length p to vertices t_h^A for all $A \in F$ and $h \in \{0, \dots, 3p - q\}$ (note that $3p - q \leq p$ as $2p \leq q$). If $3p < q$, then we connect r by paths of length $q - 2p$ to vertices $t_0^A = v_A$ for all $A \in F$. Note that in this case $q - 2p > p$ and that in both cases the distance between c and any v'_A is exactly q . Indeed, in the first case the shortest path contains vertices c, r, t_{3p-q}^A , and $t_p^A = v'_A$ and the distances are p, p , and $p - (3p - q) = q - 2p$, respectively. In the later case, the shortest path contains vertices c, r, v_A , and v'_A and the distances are $p, q - 2p$, and p , respectively. Let us denote T' the set of vertices introduced in this step. This finishes the construction of the graph G .

Let us now explain the intuition behind Construction 4.5 (cf. Figure 1). Vertex c can q -dominate all vertices in V_F, V'_F, C_r , vertices r and s and the paths connecting these. Furthermore, vertex c q -dominates the most vertices among all vertices that q -dominate s and, thus, we can assume that c is any optimal solution of (G, k') . The vertices of G that remain to be q -dominated are the vertices in C_U, V_U and the corresponding vertices in C . Our construction enforces that the vertices in V_F are the only ones suitable for that and, hence, their selection corresponds to a set cover in the original instance.

Lemma 4.6. Let (U, F, k) be an instance of SET COVER and $0 < p \leq \frac{1}{2}q$. Let (G, k') be the instance obtained from (U, F, k) by applying Construction 4.5. Then (U, F, k) is a yes-instance of SET COVER if and only if (G, k') is a yes-instance of p -SECLUDED q -DOMINATING SET.

Proof. (\Rightarrow) Let us first show that if there is $X \subseteq F$ such that $|X| \leq k$ and $\bigcup X = U$, then there is a subset S of $V(G)$ such that $V(G) = N_G^q[S]$, and $|N_G^p[S]| \leq k'$. Indeed, take a set $S = \{c\} \cup \{v_A \mid A \in X\}$. It is a routine to check that all the following vertices are in distance at most q to c and, hence, they are in $N_G^q[S]$: vertices s and r and the vertices on the paths between c and these vertices, vertices in C_r, T', T, V_F , and V'_F . For each $u \in V_U$ there is $A \in X$ such that $u \in A$ (as $\bigcup X = U$). As u is adjacent to v'_A , u is in distance (at most) $p + 1$ to v_A and, thus, it is in $N_G^q[S]$. The same holds for all vertices in C_U . Moreover, each vertex in $B \cup C$ is in distance at most $q - p - 1$ to some vertex in V_U . Therefore, all vertices in $B \cup C$ are in $N_G^q[S]$ and $V(G) = N_G^q[S]$.

It remains to bound $|N_G^p[S]|$. The p -neighborhood of c is formed by p vertices on the path to s and p vertices on the path to r (including r itself), hence it is of size exactly $2p$. For each vertex in V_F , its p -neighborhood is only formed by vertices in $T \cup T' \cup V'_F \cup \{r\}$. In particular, if $3p \geq q$, then the p -neighborhood of v_A contains p vertices of the form t_h^A for $h \in \{1, \dots, p\}$, $p - 1$ vertices on the path from t_0^A to r (excluding $v_A = t_0^A$, and r) and $p - h$ vertices on the path from t_h^A to r (excluding t_h^A) for $h \in \{1, \dots, 3p - q\}$. This, together gives

$$1 + p + \sum_{h=0}^{3p-q} (p - h) - 1 = p + \frac{1}{2}(3p - q + 1)(q - p). \quad (1)$$

If $3p < q$, then the p -neighborhood of v_A contains p vertices of the form t_h^A for $h \in \{1, \dots, p\}$ and p vertices on the path from v_A to r , that is, $2p$ vertices in total. It follows, that $|N_G^p[S]| \leq k'$.

(\Leftarrow) For the other direction, let us assume that there is a subset S of $V(G)$ such that $V(G) = N_G^q[S]$, and $|N_G^p[S]| \leq k'$.

We first observe, that $S \cap (B \cup C \cup T \cup V'_F \cup V_U \cup C_U) = \emptyset$. Notice that for every $b_h^u \in B$ we have that $\{b_h^u\} \cup C_h^u$ is a clique of size $k' + 1$ and hence, by Observation 4.2, for $v \in B \cup C$ we have that $v \notin S$. Similarly, for any $v'_A \in V'_F$ we have that $\{v'_A\} \cup C_U$ is a clique of size $k' + 1$ and furthermore we have that for each $t \in T$ there is a $v'_A \in V'_F$ such that $d_G(t, v'_A) < p$. It follows by Observation 4.2 that for every $v \in T \cup V'_F \cup C_U$ we have that $v \notin S$. For any $u \in V_U$ we have that $\{u\} \cup V_U$ is a clique of size $k' + 1$. Observation 4.2 yields that $S \cap V_U = \emptyset$. Further observe that in case $3p \geq q$ we have that $d_G(t', r) < p$ for each $t' \in T'$ and, by Observation 4.2, $S \cap T' = \emptyset$. It follows that S only contains vertices from V_F and from the path from s to c . In case $3p < q$ the set S may also contain vertices from the set T' , but notably, all these vertices are in distance at least $p + 1$ to any vertex in V_U .

Now, let us denote $X = \{A \mid v_A \in S\}$. We claim that X is a set cover for U of size at most k , that is, $|X| \leq k$ and $\bigcup X = U$. Suppose it is not a set cover. Then there is a $u \in U \setminus \bigcup X$. If $q - p = 1$, then, since S is a solution, there must be a vertex in S in distance at most q from u in G . As $V_U \cap S = \emptyset$, this vertex must be in distance at most $q - 1 = p$ from v'_A for some $A \in F$, $u \in A$. Similarly, if $q - p \geq 2$, then, since S is a solution, there must be a vertex in S in distance at most q from $v \in C_{q-p-2}^u$. As $(B \cup C \cup V_U) \cap S = \emptyset$, this vertex must be in distance at most $q - (q - p) = p$ from v'_A for some $A \in F$, $u \in A$. Since this vertex is not in $(B \cup C \cup V_U \cup T \cup V'_F)$, and if $3q \geq p$ also not in T' , it follows that this vertex is in V_F and it is the vertex v_A . But then $A \in X$ and $u \in \bigcup X$, a contradiction.

Before we show that $|X| \leq k$, observe that in S , there must be a vertex on the path between c and s (possibly one of the endpoints), this vertex is not in V_F , and has a p -neighborhood of size at least $p + 1$. For $3p < q$ each vertex of V_F has p vertices of the path to r and to the corresponding vertex of V'_F in its closed p -neighborhood. These $2p + 1$ vertices are not in p -neighborhood of any other vertex in V_F . It follows that $|X| \leq |S \cap V_F| \leq \frac{k' - (p + 1)}{2p + 1} = \frac{(k + 1) \cdot (2p + 1) - (p + 1)}{2p + 1} < k + 1$. Finally, in case $3p \geq q$, the p neighborhood of each vertex in V_F contains $p + \frac{1}{2}(3p - q + 1)(q - p)$ vertices (see Equation 1) which are not in a p -neighborhood of any other vertex in V_F . It follows that

$$\begin{aligned} |X| &= |S \cap V_F| \\ &\leq \frac{k' - (p + 1)}{p + \frac{1}{2}(3p - q + 1)(q - p)} \\ &= \frac{2p + 1 + k \cdot [p + \frac{1}{2}(3p - q + 1)(q - p)] - (p + 1)}{p + \frac{1}{2}(3p - q + 1)(q - p)} \\ &= k + \frac{p}{p + \frac{1}{2}(3p - q + 1)(q - p)} \\ &< k + 1. \end{aligned}$$

Thus in all cases $|X| \leq k$, finishing the proof of equivalence of the instances. Since the construction can be performed in polynomial time and k' is linear in k , the result follows. \square

As Construction 4.5 can be done in polynomial time, Theorem 4.4 immediately follows from Lemma 4.6. For SMALL p -SECLUDED q -DOMINATING SET, we remark that we can adapt the reduction for Theorem 4.3: instead of restricting the size of the closed neighborhood of the q -dominating set to at most $p + 1 + |F| \cdot p + k$, we restrict the size of the q -dominating set to at most $k + 1$ and the size of its open neighborhood to at most $p + |F| \cdot p$. Analogously, we can adapt the reduction for Theorem 4.4. This yields the following hardness results.

Corollary 4.7. For any $0 < p < q$, SMALL p -SECLUDED q -DOMINATING SET is NP-hard. Moreover, it does not admit a polynomial kernel with respect to $(k + \ell)$ unless $\text{NP} \subseteq \text{coNP}/\text{poly}$. For any $0 < p \leq \frac{1}{2}q$, SMALL p -SECLUDED q -DOMINATING SET is W[2]-hard with respect to $(k + \ell)$.

Now we look at the remaining choices for p and q , that is all p, q with $p > \frac{1}{2}q$. In these cases we can show fixed-parameter tractability. The crucial difference to the case $p \leq \frac{1}{2}q$ is the following. Clearly, in either case, if a solution exists, then the solution is contained in the set Y of vertices whose p -neighborhood has size at most $k + \ell$. In the case $p \leq \frac{1}{2}q$ we cannot bound the size of Y in k, ℓ or q : Indeed, in our hardness

reduction, the vertex set V_F contains an in k, ℓ , and q unbounded number of vertices, but all of these vertices have p -neighborhood of size at most $k + \ell$ and are eligible for the solution. However, in the case $p > \frac{1}{2}q$, we can show that the size of Y is bounded from above by a function in k, ℓ , and q (otherwise we can answer “no”). Roughly, this is because whenever two vertices in Y q -dominate each other, there is a path of length at most $q < 2p$ connecting them such that all of the vertices on this path have a small neighborhood. This allows us to upper-bound the number of vertices in Y that can be q -dominated by one single vertex of Y . Hence, if Y is too large we can reject, and otherwise we can brute force on the set Y of vertices whose p -neighborhood has size at most $k + \ell$ (which can be found in polynomial time) to find a solution, leading to our fixed-parameter algorithm.

Theorem 4.8. For any $p > \frac{1}{2}q$, SMALL p -SECLUDED q -DOMINATING SET can be solved in $O(mk^{k+2}(k+\ell)^{qk})$ time and, hence, it is fixed-parameter tractable with respect to $k + \ell$.

Proof. Consider a solution S for an instance (G, k, ℓ) of SMALL p -SECLUDED q -DOMINATING SET. If $x \in S$, then $|N^p[x]| \leq k + \ell$, since $|S| \leq k$ and $|N^p(S)| \leq \ell$. Moreover, $|N^p[x]| \leq k + \ell$ implies $|N[v]| \leq k + \ell$ for every $v \in N^{p-1}[x]$. It follows that, if $|N^p[y]| \leq k + \ell$ and $y \notin S$, then for each $x \in N^q[y] \cap S$ every vertex on every x - y path of length at most $2p - 1 \geq q$ has degree at most $k + \ell - 1$, since each such vertex has distance at most $p - 1$ to x or y . We point out that this property only holds if $p > q/2$, making this case different to the case $0 < p \leq q/2$ (see Corollary 4.7).

If $k + \ell = 1$, then either G has at most one vertex or (G, k, ℓ) is a no-instance. Hence, in the following, we assume $k + \ell \geq 2$. We call vertices u and v *linked*, if there is a path of length at most q between u and v in G such that the degree of every vertex on the path is at most $k + \ell - 1$. Let $B[u] = \{v \mid u \text{ and } v \text{ are linked}\}$. We claim that $|B[v]| \leq (k + \ell)^q$ for any v .

To prove the claim, let us denote $B^i[v]$ the set of vertices u such that there is a path of length at most i between u and v in G such that the degree of every vertex on the path is at most $k + \ell - 1$. Obviously, $B^q[v] = B[v]$. We prove by induction, that $|B^i[v]| \leq (k + \ell)^i$ for every $i \in \mathbb{Z}_0^+$. The claim then follows. For $i = 0$ we have $B^0[v] = \{v\}$ and $|B^0[v]| = 1 \leq (k + \ell)^0 = 1$, showing the basic step. For $i \geq 1$ and $u \in B^i[v] \setminus B^{i-1}[v]$, let p_0, p_1, \dots, p_i be the u - v path showing that $u \in B^i[v]$, i.e., $p_0 = u, p_i = v$, and for every $j \in \{0, \dots, i\}$ we have $\deg p_j \leq k + \ell - 1$. Then $p_1 \in B^{i-1}[v]$ and u is a neighbor of p_1 . Since the vertices in $B^{i-1}[v]$ have in total at most $(k + \ell - 1) \cdot |B^{i-1}[v]|$ neighbors and $|B^{i-1}[v]| \leq (k + \ell)^{i-1}$ by induction hypothesis, we have $|B^i[v]| \leq |B^{i-1}[v]| + (k + \ell - 1) \cdot |B^{i-1}[v]| = (k + \ell) \cdot |B^{i-1}[v]| \leq (k + \ell) \cdot (k + \ell)^{i-1} = (k + \ell)^i$. This gives the induction step and finishes the proof of the claim.

Let $Y = \{y \mid |N^p[y]| \leq k + \ell\}$. Obviously, we have $S \subseteq Y$, since $|N^p[S]| \leq k + \ell$. If $y \in Y \setminus S$, then there is $x \in S$ such that x and y are linked. It follows that $y \in B[x]$ and, thus, $Y \subseteq \bigcup_{x \in S} B[x]$. Hence, $|Y| \leq k \cdot (k + \ell)^q \leq (k + \ell)^{q+1}$.

This suggests the following algorithm for SMALL p -SECLUDED q -DOMINATING SET: Find the set Y . If $|Y| > k \cdot (k + \ell)^q$, then answer “no”. Otherwise, for each $k' \leq k$ and each size- k' subset S' of Y , check whether S' is a p -secluded q -dominating set in G . If any such set is found, return it. Otherwise, answer “no”. Since $S \subseteq Y$, this check is exhaustive.

As to the running time, the set Y can be determined in $O(n(k + \ell))$ time by running a BFS from each vertex and stopping it after it discovers $k + \ell$ vertices or all vertices in distance at most p , whichever occurs earlier. Then, there are $k \cdot \binom{k + \ell}{k} \leq k^{k+1} (k + \ell)^{qk}$ candidate subsets of Y . For each such set S' we can check whether it is a p -secluded q -dominating set in G by running a BFS from each vertex of S' and marking the vertices which are in distance at most p and at most q , respectively. This takes $O(mk)$ time. Hence, in total, the algorithm runs in $O(mk^{k+2}(k + \ell)^{qk})$ time. \square

By Observation 1.1, the previous result transfers to p -SECLUDED q -DOMINATING SET parameterized by k .

Corollary 4.9. For any $p > \frac{1}{2}q$, p -SECLUDED q -DOMINATING SET is fixed-parameter tractable with respect to k .

5. \mathcal{F} -free Vertex Deletion

In this section, we study the \mathcal{F} -FREE VERTEX DELETION (\mathcal{F} -FVD) problem for families \mathcal{F} of graphs with at most a constant number c of vertices, that is, the problem of destroying all induced subgraphs isomorphic to graphs in \mathcal{F} by at most k vertex deletions. The problem can, in particular, model various graph clustering tasks [4, 29], where the secluded variants can be naturally interpreted as removing a small set of outliers that are weakly connected to the clusters.

5.1. Secluded \mathcal{F} -free Vertex Deletion

In this section, we prove a polynomial-size problem kernel for SECLUDED \mathcal{F} -FREE VERTEX DELETION, where \mathcal{F} is a family of graphs with at most a constant number c of vertices:

Problem (SECLUDED \mathcal{F} -FREE VERTEX DELETION).

Input: A graph $G = (V, E)$ and an integer k .

Question: Is there a set $S \subseteq V$ such that $G - S$ is \mathcal{F} -free and $|N_G[S]| \leq k$?

Henceforth, we call a set $S \subseteq V$ such that $G - S$ is \mathcal{F} -free an \mathcal{F} -free vertex deletion set.

Note that SECLUDED \mathcal{F} -FREE VERTEX DELETION can be polynomial-time solvable for some families \mathcal{F} for which \mathcal{F} -FVD is NP-hard: VERTEX COVER (where \mathcal{F} contains only the graph consisting of a single edge) is NP-hard, yet any vertex cover S satisfies $N[S] = V$. Therefore, an instance to SECLUDED VERTEX COVER is a yes-instance if and only if $k \geq n$. In general, however, one can show that SECLUDED \mathcal{F} -FREE VERTEX DELETION is NP-complete for every family \mathcal{F} that includes only graphs of minimum vertex degree two (Theorem 5.1). We mention in passing that, from this peculiar difference of the complexity of VERTEX COVER and SECLUDED VERTEX COVER, it would be interesting to find properties of \mathcal{F} which govern whether SECLUDED \mathcal{F} -FREE VERTEX DELETION is NP-hard or polynomial-time solvable along the lines of the well-known dichotomy results [21, 35].

Theorem 5.1. For each family \mathcal{F} containing only graphs of minimum vertex degree two, SECLUDED \mathcal{F} -FREE VERTEX DELETION is NP-complete.

Proof. We reduce from \mathcal{F} -FREE VERTEX DELETION, which is NP-complete for all \mathcal{F} that contain only graphs of minimum vertex degree two [35]. Given an instance (G, s) of \mathcal{F} -FREE VERTEX DELETION where G contains n vertices, we add $n + 1$ new degree-one neighbors to each vertex in G . In this way, we obtain an instance (G', k) of SECLUDED \mathcal{F} -FREE VERTEX DELETION by setting $k = s \cdot (n + 1) + n$.

Clearly, each \mathcal{F} -free vertex deletion set S of size at most s for G is a \mathcal{F} -free vertex deletion set for G' as each graph in \mathcal{F} contains no degree-one vertices. Furthermore, clearly, $|N_{G'}[S]| \leq s \cdot (n + 1) + n = k$.

In the other direction, for each \mathcal{F} -free vertex deletion set S for G' we may assume that no degree-one vertex is contained in S , hence, S is a \mathcal{F} -free vertex deletion set for G . Furthermore, as each vertex in $V(G)$ incurs at least $n + 1$ vertices in the closed neighborhood $N_{G'}(S)$, if this neighborhood has size at most k , then there are at most s vertices in S . \square

It is easy to see that SECLUDED \mathcal{F} -FREE VERTEX DELETION is fixed-parameter tractable. More specifically, it is solvable in $c^k \cdot \text{poly}(n)$ time: simply enumerate all inclusion-minimal \mathcal{F} -free vertex deletion sets S of size at most k using the standard search tree algorithm described by Cai [9] and check $|N[S]| \leq k$ for each of them. This works because, for any \mathcal{F} -free vertex deletion set S with $|N[S]| \leq k$, we can assume that S is an inclusion-minimal \mathcal{F} -free vertex deletion set since $|N[S']| \leq |N[S]|$ for every $S' \subsetneq S$.

We complement this observation of fixed-parameter tractability by the following kernelization result.

Theorem 5.2. SECLUDED \mathcal{F} -FREE VERTEX DELETION has a problem kernel comprising $O(k^{c+1})$ vertices, where c is the maximum number of vertices in any graph of \mathcal{F} .

Our proof of Theorem 5.2 exploits *expressive* kernelization algorithms for d -HITTING SET [2, 3, 19], which preserve inclusion-minimal solutions and that return subgraphs of the input hypergraph as kernels: Herein, given a hypergraph $H = (U, \mathcal{C})$ with $|C| \leq d$ for each $C \in \mathcal{C}$, and an integer k , d -HITTING SET asks whether

there is a *hitting set* $S \subseteq U$ with $|S| \leq k$, that is, $C \cap S \neq \emptyset$ for each $C \in \mathcal{C}$. Our kernelization for SECLUDED \mathcal{F} -FREE VERTEX DELETION is based on transforming the input instance (G, k) to a d -HITTING SET instance (H, k) , computing an expressive d -HITTING SET problem kernel (H', k) , and outputting a SECLUDED \mathcal{F} -FREE VERTEX DELETION instance (G', k) , where G' is the graph induced by the vertices remaining in H' together with at most $k + 1$ additional neighbors for each vertex in G .

Definition 5.3. Let $(G = (V, E), k)$ be an instance of SECLUDED \mathcal{F} -FREE VERTEX DELETION. For a vertex $v \in V$, let $N_j(v) \subseteq N_G(v)$ be a set of j arbitrary neighbors of v , or $N_j(v) := N_G(v)$ if v has degree less than j . For a subset $S \subseteq V$, let $N_j(S) := \bigcup_{v \in S} N_j(v)$. Moreover, let

$c := \max_{F \in \mathcal{F}} |V(F)|$ be the maximum number of vertices in any graph in \mathcal{F} ,

$H = (U, \mathcal{C})$ be the hypergraph with $U := V$ and $\mathcal{C} := \{S \subseteq V \mid G[S] \in \mathcal{F}\}$,

$H' = (U', \mathcal{C}')$ be a subgraph of H with $|U'| \in O(k^c)$ such that each set $S \subseteq U$ with $|S| \leq k$ is an inclusion-minimal hitting set for H if and only if it is for H' , and

$G' = (V', E')$ be the subgraph of G induced by $U' \cup N_{k+1}(U')$.

To prove Theorem 5.2, we show that (G', k) is a problem kernel for the input instance (G, k) . The subgraph H' exists and is computable in linear time from H [3, 19]. Moreover, for constant c , one can compute H from G and G' from H' in polynomial time. It is obvious that the number of vertices of G' is $O(k^{c+1})$. Hence, it remains to show that (G', k) is a yes-instance if and only if (G, k) is. This is achieved by the following two lemmas.

Lemma 5.4. For any $S \subseteq U'$ with $|N_{G'}[S]| \leq k$, it holds that $N_G[S] = N_{G'}[S]$.

Proof. Since $S \subseteq U' \subseteq V' \cap V$ and since G' is a subgraph of G , it is clear that $N_G[S] \supseteq N_{G'}[S]$. For the opposite direction, observe that each $v \in S$ has degree at most k in G' . Thus, v has degree at most k in G since, otherwise, $k + 1$ of its neighbors would be in G' by construction. Thus, $N_{G'}(v) \supseteq N_{k+1}(v) = N_G(v)$ for all $v \in S$ and, thus, $N_{G'}[S] \supseteq N_G[S]$. \square

Lemma 5.5. Graph G allows for an \mathcal{F} -free vertex deletion set S with $|N_G[S]| \leq k$ if and only if G' allows for an \mathcal{F} -free vertex deletion set S with $|N_{G'}[S]| \leq k$.

Proof. Let S be an inclusion-minimal \mathcal{F} -free vertex deletion set with $|N_G[S]| \leq k$ for G . Then S is an inclusion-minimal hitting set for H and, by construction, also for H' . Thus, S consists only of vertices of G' . Since G' is an induced subgraph of G , it holds that $G' - S$ is an induced subgraph of $G - S$, which is \mathcal{F} -free. Thus, $G' - S$ is also \mathcal{F} -free and S is an \mathcal{F} -free vertex deletion set for G' . Moreover, $|N_{G'}[S]| \leq |N_G[S]| \leq k$ follows since G' is a subgraph of G .

Now, let S be an \mathcal{F} -free vertex deletion set with $|N_{G'}[S]| \leq k$ for G' . Then $S \cap U'$ is a hitting set for H' : if there was a set $C \in \mathcal{C}'$ with $C \cap S = \emptyset$, then, by construction of H' and G' , $G'[C] = G[C] \in \mathcal{F}$ would remain a forbidden induced subgraph in $G' - S$. Thus, S contains an inclusion-minimal hitting set $S' \subseteq U'$ for H' . Since $|S'| \leq k$, it is, by construction of H' , also a hitting set for H . Now, by construction of H from G , S' is a \mathcal{F} -free vertex deletion set for G . Finally, since $|N_{G'}[S']| \leq |N_{G'}[S]| \leq k$, we also have $|N_G[S']| = |N_{G'}[S']| \leq k$ by Lemma 5.4. \square

5.2. Small Secluded \mathcal{F} -free Vertex Deletion

In this subsection, we present a fixed-parameter algorithm for the following problem parameterized by $\ell + k$.

Problem (SMALL SECLUDED \mathcal{F} -FREE VERTEX DELETION).

Input: A graph $G = (V, E)$ and two integers k, ℓ .

Question: Is there a subset $S \subseteq V$ such that $G - S$ is \mathcal{F} -free, $|S| \leq k$, and $|N_G(S)| \leq \ell$?

As before, we call a set $S \subseteq V$ such that $G - S$ is \mathcal{F} -free an \mathcal{F} -free vertex deletion set.

In the previous section, we discussed a simple search tree algorithm for SECLUDED \mathcal{F} -FREE VERTEX DELETION that was based on the fact that we could assume that our solution is an inclusion-minimal \mathcal{F} -free vertex deletion set. However, an \mathcal{F} -free vertex deletion set S with $|S| \leq k$ and $|N_G(S)| \leq \ell$ is not necessarily inclusion-minimal: some vertices may have been added to S just in order to shrink its open neighborhood. However, the following simple lemma limits the number of possible candidate vertices that can be used to enlarge S in order to shrink $N(S)$, which we will use in a branching algorithm.

Lemma 5.6. Let S be an \mathcal{F} -free vertex deletion set and $S' \supseteq S$ such that $|S'| \leq k$ and $|N_G(S')| \leq \ell$, then $|N_G(S)| \leq \ell + k$.

Proof. $|N_G(S)| = |N_G[S] \setminus S| \leq |N_G[S'] \setminus S| \leq |N_G[S']| \leq |N_G(S') \cup S'| \leq \ell + k$. \square

Theorem 5.7. SMALL SECLUDED \mathcal{F} -FREE VERTEX DELETION can be solved in $\max\{c, k + \ell\}^k \cdot \text{poly}(n)$ -time, where c is the maximum number of vertices in any graph of \mathcal{F} .

Proof. First, enumerate all inclusion-minimal \mathcal{F} -free vertex deletion sets S with $|S| \leq k$. This is possible in $c^k \cdot \text{poly}(n)$ time using the generic search tree algorithm described by Cai [9]. For each $k' \leq k$, this search tree algorithm generates at most $c^{k'}$ sets of size k' . For each enumerated set S of k' elements, do the following:

1. If $|N_G(S)| \leq \ell$, then output S as our solution.
2. If $|N_G(S)| > \ell + k$, then S cannot be part of a solution S' with $|N_G(S')| \leq \ell$ by Lemma 5.6, we proceed with the next set.
3. Otherwise, initiate a recursive branching: recursively branch into at most $\ell + k$ possibilities of adding a vertex from $N_G(S)$ to S as long as $|S| \leq k$.

The recursive branching initiated at step 3 stops at depth $k - k'$ since, after adding $k - k'$ vertices to S , one obtains a set of size k . Hence, the total running time of our algorithm is

$$\text{poly}(n) \cdot \sum_{k'=1}^k c^{k'} (\ell + k)^{k-k'} = \text{poly}(n) \cdot \sum_{k'=1}^k \max\{c, \ell + k\}^k = \text{poly}(n) \cdot \max\{c, \ell + k\}^k. \quad \square$$

Given Theorem 5.7, a natural question is whether the problem allows for a polynomial kernel.

6. Feedback Vertex Set

In this section, we study secluded versions of the FEEDBACK VERTEX SET (FVS) problem, which asks, given a graph G and an integer k , whether there is a set $W \subseteq V(G)$, $|W| \leq k$, such that $G - W$ is cycle-free.

6.1. Secluded Feedback Vertex Set

We show in this subsection that the problem below is NP-hard and admits a polynomial kernel.

Problem (SECLUDED FEEDBACK VERTEX SET (SFVS)).

Input: A graph $G = (V, E)$ and an integer k .

Question: Is there a set $S \subseteq V$ such that $G - S$ is cycle-free and $|N_G[S]| \leq k$?

Theorem 6.1. SECLUDED FEEDBACK VERTEX SET is NP-hard.

The proof is by a reduction from the FVS problem and works by attaching to each vertex in the original graph a large set of new degree-one neighbors.

Proof. We provide a polynomial time many-one reduction from FEEDBACK VERTEX SET. Let $(G = (V, E), k)$ be an instance of FEEDBACK VERTEX SET. We construct an equivalent instance $(G' = (V', E'), k')$ of SFVS as follows. To obtain G' , for each vertex $v \in V(G)$ add n^2 vertices and connect them to v . Observe that the *added vertices* have degree one and thus are never part of a cycle in G' . Further, set $k' = k \cdot (n^2 + n)$. We claim that (G, k) is a yes-instance of FVS if and only (G', k') is a yes-instance of SFVS.

(\Rightarrow) Let $S \subseteq V(G)$ be a feedback vertex set in G . Then the corresponding vertices in G' form a feedback vertex set in G' . Moreover, we have k vertices, each having at most $n^2 + n$ neighbors. Thus, $|N_{G'}[S]| \leq k \cdot (n^2 + n) = k'$. It follows that (G', k') is a yes-instance of SFVS.

(\Leftarrow) Conversely, let S be a minimal solution to (G', k') , that is, S is a feedback vertex set in G' such that $|N_{G'}[S]| \leq k'$ and $S \setminus \{v\}$ is not a feedback vertex set in G' for every $v \in S$. By minimality of S , and since the added vertices do not appear in any cycle in G' , S does not contain any of the added vertices. Hence $S \subseteq V$ and, thus $|S| \leq k$ as each vertex in V has at least n^2 private neighbors in G' . Thus, since S forms a feedback vertex set in G' , S also forms a feedback vertex set in G . It follows that (G, k) is a yes-instance of FVS. \square

On the positive side, SFVS remains fixed-parameter tractable with respect to k :

Theorem 6.2. SECLUDED FEEDBACK VERTEX SET admits a kernel with $O(k^5)$ vertices.

In the remainder of this section, we describe the data reduction rules that yield the polynomial-size problem kernel. The reduction rules are inspired by the kernelization algorithm for the TREE DELETION SET problem given by Giannopoulou et al. [27].

We start by introducing the following notation. A *2-core* [41] of a graph G is a maximum subgraph H of G such that, for each $v \in V(H)$, we have $\deg_H(v) \geq 2$. Note that a 2-core H of a given graph G is unique and can be found in polynomial time [41]. If H is a 2-core of G , then we use $\deg_{H|0}(v)$ to denote $\deg_H(v)$ if $v \in V(H)$ and $\deg_{H|0}(v) = 0$ if $v \notin V(H)$.

Observation 6.3. Let G be a graph, H its 2-core, and C a connected component of $G - V(H)$. Then $|N(C) \cap V(H)| \leq 1$ and $|N(H) \cap V(C)| \leq 1$.

Proof. We only show the first statement. The second statement follows analogously. Towards a contradiction, assume that $|N(C) \cap V(H)| \geq 2$. Then, there are vertices $x, y \in V(H)$ with $x \neq y$ such that x and y have neighbors $a, b \in V(C)$. If $a = b$, then $G' = G[V(H) \cup \{a\}]$ is a subgraph of G such that $\deg_{G'}(v) \geq 2$ for every $v \in V(G')$, contradicting the choice of H as the 2-core of G . If $a \neq b$, then, since C is connected, there is a path P_C in C connecting a and b . Thus, $G' = G[V(H) \cup V(P_C)]$ is a subgraph of G such that $\deg_{G'}(v) \geq 2$ for every $v \in V(G')$, again contradicting the choice of H as the 2-core of G . \square

Note that only the vertices in the 2-core are involved in cycles of G . However, the vertices outside the 2-core can influence the size of the closed neighborhood of the feedback vertex set. Next, we apply the following reduction rules to our input instance with G given its 2-core H .

We say that a feedback vertex set F in G is *secluded* if $|N[F]| \leq k$. Further, we say that a secluded feedback vertex set F in G is *minimal*, if $F \setminus \{v\}$ is not a secluded feedback vertex set in G for all $v \in F$.

Reduction Rule 6.4. If $\deg_{H|0}(v) = 0$ for every $v \in N[u]$, then delete u .

Proof of correctness. Let F be a minimal secluded feedback vertex set in G . Since $\deg_{H|0}(v) = 0$ for all $v \in N[u]$, none of them is involved in a cycle. Hence, $N[u] \cap F = \emptyset$. In particular, it follows from $N(u) \cap F = \emptyset$ that $u \notin N[F]$. Hence, F is a secluded feedback vertex set in $G - \{u\}$ as well.

Conversely, let F be a minimal secluded feedback vertex set in $G_u := G - \{u\}$. We have to show that F is a secluded feedback vertex set in G as well. First observe that since $\deg_{H|0}(v) = 0$ for all $v \in N_G[u]$, H is also the 2-core of G_u . As only vertices in H participate in cycles of G_u and F is chosen as minimal, none of the vertices $N_G(u) \subseteq V(G_u)$ is contained in F . It follows that $|N_G[F]| = |N_{G_u}[F]| \leq k$, and thus F is a secluded feedback vertex set in G as well. \square

Note that, if Reduction Rule 6.4 has been exhaustively applied, then $\deg_{H|0}(v) = 0$ implies that v has exactly one neighbor, which is in the 2-core of the graph.

Reduction Rule 6.5. If $v_0, v_1, \dots, v_\ell, v_{\ell+1}$ is a path in the input graph such that $\ell \geq 3$, $\deg_{H|_0}(v_i) = 2$ for every $i \in \{1, \dots, \ell\}$, $\deg_{H|_0}(v_0) \geq 2$, and $\deg_{H|_0}(v_{\ell+1}) \geq 2$, then let $r = \min\{\deg_G(v_i) \mid i \in \{1, \dots, \ell\}\} - 2$ and remove vertices v_1, \dots, v_ℓ and their neighbors not in the 2-core. Then introduce two new vertices u_1 and u_2 with edges $\{v_0, u_1\}$, $\{u_1, u_2\}$, and $\{u_2, v_{\ell+1}\}$ and $2r$ further new vertices and connect u_1 with r of them and u_2 with another r of them.

Proof of correctness. Let F be a minimal secluded feedback vertex set in G , and let G' be the graph obtained from G by applying Reduction Rule 6.5. Suppose $F \cap \{v_1, \dots, v_\ell\} \neq \emptyset$. Since $\deg_{H|_0}(v_i) = 2$ for all $i \in [\ell]$, each of the vertices v_1, \dots, v_ℓ participates in the same set of cycles of G . Hence, it follows that $F \cap \{v_1, \dots, v_\ell\} = \{v_q\}$ for some $q \in [\ell]$. Moreover, the set of cycles where v_1, \dots, v_ℓ appear in is a subset of the set of cycles where v_0 appears in and a subset of the set of cycles where $v_{\ell+1}$ appears in. Hence, due to minimality of F we have that $v_q \in F$ implies $v_0 \notin F$ and $v_{\ell+1} \notin F$. Due to the definition of r the number of neighbors of v_q not in the 2-core is at least r . Then $F' = (F \setminus \{v_q\}) \cup \{u_1\}$ is a secluded feedback vertex set of G' with $|F'| = |F|$ and $|N_G[F]| \geq |N_{G'}[F']|$.

Suppose $F \cap \{v_1, \dots, v_\ell\} = \emptyset$ but $F \cap \{v_0, v_{\ell+1}\} \neq \emptyset$. Then $|F \cap \{v_0, v_{\ell+1}\}| = |N_G[F] \cap \{v_1, v_\ell\}| = |N_{G'}[F] \cap \{u_1, u_2\}|$. It follows that F is a secluded feedback vertex set in G' with $|N_{G'}[F]| = |N_G[F]|$.

The case where $F \cap \{v_0, \dots, v_{\ell+1}\} = \emptyset$ is trivial.

Conversely, let F be a minimal secluded feedback vertex set in G' . Suppose that $F \cap \{u_1, u_2\} \neq \emptyset$. Since F is minimal, either u_1 or u_2 is contained in F , since both vertices participate in the same set of cycles in G' . Without loss of generality, let $u_1 \in F$. Moreover, $F \cap \{v_0, v_\ell\} = \emptyset$, as otherwise $F \setminus \{u_1\}$ is a smaller secluded feedback vertex set in G' , contradicting the minimality of F . By the choice of r , there exists $q \in [\ell]$ such that $\deg_G(v_q) - 2 = r$. Then $F' := (F \setminus \{u_1\}) \cup \{v_q\}$ is a feedback vertex set in G with $|N_G[F']| = |N_{G'}[F]|$.

Suppose that $F \cap \{v_0, v_{\ell+1}\} \neq \emptyset$. Since F is minimal, it follows that $F \cap \{u_1, u_2\} = \emptyset$. Observe that F is also a feedback vertex set in G , as v_0 and $v_{\ell+1}$ participate in each cycle containing any vertex in $\{v_1, \dots, v_\ell\}$. Since $|F \cap \{v_0, v_{\ell+1}\}| = |N_{G'}[F] \cap \{u_1, u_2\}| = |N_G[F] \cap \{v_1, v_\ell\}|$, it follows that $|N_G[F]| = |N_{G'}[F]|$. Hence, F is a secluded feedback vertex set in G .

The case where $F \cap \{v_0, u_1, u_2, v_{\ell+1}\} = \emptyset$ is trivial. \square

For $x \in V(G)$, we denote by $\text{petal}(x)$ the maximum number of cycles only intersecting in x .

Reduction Rule 6.6. If there is a vertex $x \in V(G)$ such that $\text{petal}(x) \geq \lceil \frac{k}{2} \rceil$, then output that (G, k) is a no-instance of SFVS.

Proof of correctness. There are at least $\lceil \frac{k}{2} \rceil$ cycles in G , which are vertex-disjoint except for x . Assume that G allows a feedback vertex set F with $|N_G[F]| \leq k$. Clearly, F must contain at least one vertex in each of the cycles. Therefore $N[F]$ must contain at least three vertices of each cycle. As only x can be shared among these triples, we get $|N_G[F]| \geq 2 \cdot \lceil \frac{k}{2} \rceil + 1 > k$. It follows that G does not admit a secluded feedback vertex set. \square

Reduction Rule 6.7. If $v \in V(G)$ is a vertex such that $\deg_G(v) > k$, but $\deg_{H|_0}(v) < \deg_G(v)$, then remove one of its neighbors not in the 2-core.

Proof of correctness. First observe that, as $\deg_G(v) > k$, vertex v cannot be contained in any secluded feedback vertex set. As additionally $\deg_{H|_0}(v) < \deg_G(v)$, we know that there is a vertex $w \in N(v) \setminus V(H)$. Since w is not in the 2-core, it is not involved in the cycles of G . Since $\deg_G(v) > k$, removing w from G results in $\deg_{G-\{w\}}(v) \geq k$ and hence, v cannot be contained in any secluded feedback vertex set of $G - \{w\}$. Altogether, G has a feedback vertex set F with $|N_G[F]| \leq k$ if and only if $G - \{w\}$ has a feedback vertex set F' with $|N_{G-\{w\}}[F']| \leq k$. \square

Reduction Rule 6.8. Let x, y be two vertices of G . If there are at least k internally vertex disjoint paths of length at least 2 between x and y in G , then output that (G, k) is a no-instance of SFVS.

Proof of correctness. Observe then that if neither x nor y belong to a feedback vertex set D of G we need at least $k - 1$ vertices to hit all the cycles, since otherwise there are at least two distinct paths P_1, P_2 of length at least 2 between x and y with $(V(P_1) \cup V(P_2)) \cap D = \emptyset$ and thus the graph induced by $V(P_1) \cup V(P_2) \cup \{x, y\}$

contains a cycle. Since each of the $k - 1$ vertices has at least two vertices in its open neighborhood and only the vertices x and y can be shared among these, the closed neighborhood contains at least $k + 1$ vertices. On the other hand, the open neighborhood of both x and y contains one vertex from each of the k paths. Therefore, their closed neighborhood is of size at least $k + 1$ and they cannot be included in the solution. \square

Note that Reduction Rules 6.4, 6.5, 6.7 and 6.8 can be applied trivially in polynomial time. Reduction Rule 6.6 can be applied exhaustively in polynomial time due to the following.

Proposition 6.9 ([42]). Let G be a graph and x be a vertex of G . In polynomial time we can either find a set of $\ell + 1$ cycles only intersecting in x (proving that $\text{petal}(x) \geq \ell + 1$) or a set of vertices $Z \subseteq V(G) \setminus \{x\}$ of size at most 2ℓ intersecting every cycle containing x .

An instance (G, k) of SFVS is called *reduced* if none of the Reduction Rules 6.4 to 6.8 can be applied. Following the proof by Giannopoulou et al. [27], we first give structural decomposition lemma, then bound the size of components of the decomposition, and finally bound the number of components in the decomposition to obtain the polynomial kernel for SFVS parameterized by k . We start with the following structural decomposition lemma, which identifies the set B .

Lemma 6.10. There is a polynomial time algorithm that given a reduced instance (G, k) of SFVS either correctly decides that (G, k) is a no-instance or finds two sets F and M' such that, denoting $B = F \cup M'$, the following holds.

- (i) F is a feedback vertex set of G .
- (ii) Each connected component of $G - B$ has at most 2 neighbors in M' .
- (iii) For every connected component C in $G - B$ and $x \in B$, $|N_G(x) \cap C| \leq 1$, that is, every vertex y of F and every vertex x of M' have at most one neighbor in every connected component C of $G - B$.
- (iv) $|B| \leq 4k^2 + 2k$.

Similar to Giannopoulou et al. [27], we also make use of the following concept. For a rooted tree T and vertex set M in $V(T)$ the lowest common ancestor-closure (*LCA-closure*) $\text{lca}(M)$ is obtained by the following process. Initially, set $M' = M$. Then, as long as there are vertices x and y in M' whose lowest common ancestor w is not in M' , add w to M' . Finally, output M' as the LCA-closure of M .

Lemma 6.11 (Fomin et al. [23]). Let T be a tree and $M \subseteq V(T)$. If $M' = \text{lca}(M)$ then $|M'| \leq 2|M|$ and for every connected component C of $T - M'$, $|N_T(C)| \leq 2$.

We continue with proving our structural decomposition lemma.

Proof of Lemma 6.10. Note that if there is a feedback vertex set of G with closed neighborhood of size at most k , then it is also a feedback vertex set in G of size at most k . Thus, we can apply the 2-approximation algorithm for FEEDBACK VERTEX SET on G due to Bafna et al. [1] to find in polynomial time a feedback vertex set F of G . If $|F| > 2k$, then we output that (G, k) is a no-instance of SFVS. Hence, we assume $|F| \leq 2k$ in the following. Since F is a feedback vertex set in G , property (i) is trivially fulfilled. Moreover, $G - F$ is a collection of trees T_1, \dots, T_ℓ . We select for each of the trees T_i some root vertex $v_i \in V(T_i)$. It remains to construct the set M' such that $F \cup M'$ fulfills conditions (ii)–(iv).

Recall that the instance (G, k) is reduced. Hence, Reduction Rule 6.6 is not applicable, and hence $\text{petal}(x) < \lceil \frac{k}{2} \rceil$ for all $x \in F$. We apply Proposition 6.9 to each vertex in $v \in F$, obtaining a set $Z_v \subseteq V(G) \setminus \{v\}$ intersecting each cycle containing v with $|Z_v| \leq k$. Let $Z := Z_1 \cup \dots \cup Z_{|F|}$ denote the union of these sets. Observe that $|Z| \leq 2k^2$. We set $M_i := T_i \cap Z$ and $M'_i := \text{lca}(M_i)$ for all $i \in [\ell]$. Observe that, by Lemma 6.11, $|M'_i| \leq 2|M_i|$. Finally, we set $M' = \bigcup_{i \in [\ell]} M'_i$ and $B = F \cup M'$ (note that $F \cap M' = \emptyset$). Observe that $|M'| \leq \sum_{i \in [\ell]} |M'_i| \leq \sum_{i \in [\ell]} 2|M_i| \leq 2|Z| \leq 4k^2$ and by Lemma 6.11, for every connected component C in $G - B$ it holds that $|N_{G-F}(C)| \leq 2$ (hence, property (ii) is fulfilled). Altogether, $|B| = |F| + |M'| \leq 2k + 4k^2$, yielding property (iv). It remains to show that property (iii) is fulfilled.

Let C be a connected component of $G - B$ and $x \in B$ some vertex. Suppose that x has two neighbors in C . Then $C_x := C \cup \{x\}$ induces a cycle in G as C is connected. If $x \in F$, then this contradicts the set $Z_x \subseteq Z \subseteq M' \cup (F \setminus \{x\})$ hitting every cycle containing x . If $x \in M'$, then this contradicts the set F hitting each cycle in G . Hence, property (iii) is fulfilled. \square

Next, we show that if B is as in Lemma 6.10, then the size and the number of the connected components in the $G - B$ is polynomially bounded from above in the size k of the closed neighborhood of the feedback vertex set in question. We first bound from above the size of the connected components in $G - B$ as follows.

Lemma 6.12. Let (G, k) and B be as in Lemma 6.10, and let C be a connected component of $G - B$. Then the number of vertices $|V(C)|$ of the connected component C is at most $(12k + 7)(k + 1)$.

Proof. Let H be the 2-core of G . We distinguish two cases on the size of $C_H := V(C) \cap V(H)$, namely $|C_H| = 0$ on the one hand, and $|C_H| > 0$ on the other hand.

Case $|C_H| = 0$: Observe that C is a connected component in $G - V(H)$. Hence, by Observation 6.3, there is at most one vertex in C adjacent to H . If $x \in V(C)$ is adjacent to H , no other vertex of C is adjacent to H . Suppose that $|V(C)| > 1$. Since C is connected, there is a vertex $u \in V(C)$ such that $N[u] \subseteq G - V(H)$. Existence of such vertex would contradict the instance being reduced with respect to Reduction Rule 6.4. Hence, $|V(C)| \leq 1$.

Case $|C_H| > 0$: Recall that (G, k) is reduced. On the one hand, due to Reduction Rule 6.4, we know that every vertex in $C - V(H)$ has a neighbor in C_H . On the other hand, due to Reduction Rule 6.7, each vertex in C_H has at most k neighbors in $C - V(H)$. Hence, it follows that $|V(C)| \leq (k + 1) \cdot |C_H|$. Consequently, it remains to bound the number of vertices in C_H .

In the following we count the number of vertices in $G[C_H]$ having degree 1, 2, and at least 3 in $G[C_H]$. Let $D_H^1 \subseteq C_H$ be the set of vertices in $G[C_H]$ having degree exactly one. Since $D_H^1 \subseteq V(H)$, it holds that $\deg_{H|0}(v) \geq 2$ for each $v \in D_H^1$. Since there is exactly one neighbor of v in $G[C_H]$, at least one other neighbor is contained in $V(H) \cap B$. Let B_C denote the vertices of C having at least one neighbor in B . Note that $D_H^1 \subseteq B_C$. Due to Lemma 6.10(ii), C has at most two neighbors in M' (recall $B = F \cup M'$). Moreover, due to Lemma 6.10(iii), each vertex in B has at most one neighbor in C . It follows that $|B_C| \leq |F| + 2 \leq 2k + 2$, and hence $|D_H^1| \leq 2k + 2$.

Let $D_H^{\geq 3} \subseteq C_H$ be the set of vertices in $G[C_H]$ having degree at least three. Since $G[C_H]$ is acyclic (recall that $F \subseteq B$ is a feedback vertex set), it follows that $D_H^{\geq 3}$ forms the leaves in $G[C_H]$. A basic observation on trees is that the number of inner vertices of degree at least three is at most the number of leaves minus one. Hence, $|D_H^{\geq 3}| \leq |D_H^1| - 1 \leq 2k + 1$.

Let $D_H^{-2} := B_C \cup D_H^{\geq 3}$. Observe that $C_H \setminus D_H^{-2}$ only contains vertices having degree exactly two in $G[C_H]$. Moreover, these vertices participate only in paths connecting vertices in D_H^{-2} . Since $|D_H^{-2}| \leq 2k + 2 + 2k + 1 = 4k + 3$, and $G[C_H]$ is acyclic, there are at most $4k + 3 - 1 = 4k + 2$ many of these paths. Moreover, due to Reduction Rule 6.5, these paths contain at most two vertices not being the endpoints. Hence, $|C_H| \leq |C_H \setminus D_H^{-2}| + |D_H^{-2}| \leq 2 \cdot (4k + 2) + 4k + 3 = 12k + 7$. It follows that $|V(C)| \leq (k + 1) \cdot |C_H| \leq (k + 1) \cdot (12k + 7)$. \square

Having an upper bound on the sizes of the set B and of each connected component in $G - B$, it remains to count the number of connected components in $G - B$.

Remark. It is easy to polynomially upper-bound the number of connected components in $G - B$. To this end, first observe that by Reduction Rule 6.4, each connected component in $G - B$ has at least one neighbor in B . Next, consider those connected components in $G - B$ having exactly one neighbor in B . Due to Reduction Rule 6.6, each vertex in B is incident to at most k connected components in $G - B$ having exactly one neighbor in B . Hence, the number of these connected components in $G - B$ is upper bounded by $|B| \cdot (k + 1)$. Last, consider those connected components in $G - B$ having at least two neighbors in B . Then it follows from Reduction Rule 6.8 that two vertices in B are together contained in the neighborhood of at most k connected components in $G - B$. Indeed, each connected component C of $G - B$ with $\{x, y\} \subseteq N_G(C)$ provides a separate path between x and y . Altogether, the number of connected components in $G - B$ is

upper bounded by $|B|^2 \cdot (k+1) \in O(k^5)$. Hence, together with Lemma 6.12, we obtain a polynomial kernel of size $O(k^7)$ for SFVS.

With the next lemma, we give an $O(k^3)$ upper bound on the number of connected components in $G - B$.

Lemma 6.13. Let (G, k) and B be as in Lemma 6.10. Then the number of connected components in $G - B$ is at most $15k^3 + 8k^2 - k - 1$.

Proof. We partition the connected components of $G - B$ by the number of their neighbors in B , namely having exactly one neighbor and having at least two neighbors in B . For $x \in B$, denote by B_x , the set of connected components in $G - B$ having vertex x as their only neighbor in B . Further, for $x, y \in B$, denote by B_{xy} , the set of connected components having at least x and y as their neighbors in B . Observe that the connected components of $G - B$ are exactly $\bigcup_{\{x,y\} \subseteq B} (B_x \cup B_{xy})$, and hence the number of the connected components of $G - B$ is at most $|\bigcup_{x \in B} B_x| + |\bigcup_{\{x,y\} \subseteq B} B_{xy}|$. Further observe that $|\bigcup_{x \in B} B_x| \leq |B|k \leq 4k^3 + 2k^2$. Hence, it remains to upper-bound the cardinality of $\bigcup_{\{x,y\} \subseteq B} (B_{xy})$. To this end, observe that

$$\bigcup_{\{x,y\} \subseteq B} (B_{xy}) = \underbrace{\bigcup_{\{x,y\} \subseteq F} (B_{xy})}_{:=B^1} \cup \underbrace{\bigcup_{x \in F, y \in M'} (B_{xy})}_{:=\tilde{B}^2} \cup \underbrace{\bigcup_{\{x,y\} \subseteq M'} (B_{xy})}_{:=B^3}. \quad (2)$$

Notice that the equality is still true if we replace \tilde{B}^2 by $B^2 := \tilde{B}^2 \setminus B^3$, since B^3 appears in the union on the right hand-side. Hence, in the remainder of this proof, we upper-bound the size of the sets B^1 , B^2 , and B^3 . Observe that the size of B^1 is upper bounded by $\binom{2k}{2}(k+1) = 2k^3 + k^2 - k$.

Next, we upper-bound the size of B^2 . To this end, let $x \in F$ be arbitrary but fixed. Consider the set S_x of vertices in M' such that there are at least two connected components of $G - B$ neighboring with both x and y . Observe that for each $y \in S_x$, the set of connected components in B^2 neighboring with both x and y is unique, as otherwise there is a connected component in B^2 containing two vertices in M' and hence belonging to B^3 , contradicting our definition of $B^2 := \tilde{B}^2 \setminus B^3$. Since for each $y \in S_x$ there are at least two connected components in B^2 , they together with x and y form a cycle in G . Hence, due to Reduction Rule 6.6, the number of vertices in S_x is at most $k/2$. On the other hand, there are at most k connected components neighboring with both x and y for any $y \in S_x$ due to Reduction Rule 6.8, since each such component provides a separate path of length at least 2 between x and y . Finally, observe that the number of vertices $y \in M'$ such that there is at most one connected component of $G - B$ neighboring with both x and y is trivially bounded by $|M'| \leq 4k^2$. Altogether, we obtain that $|B^2| \leq \sum_{x \in F} (4k^2 + (k/2)(k+1)) \leq 2k(4k^2 + (k/2)(k+1)) = 9k^3 + k^2$.

Last, we upper-bound the size of $B^3 = \bigcup_{\{x,y\} \subseteq M'}$. Observe that due to Lemma 6.10(ii), for each $x, y \in M'$ each connected component C in B_{xy} only neighbors with x and y out of M' , that is, $N(C) \cap M' = \{x, y\}$. Moreover, by the connectedness of C , x and y are connected via a path through C . By known facts on forests and trees, we know that if there are at least r paths connecting vertex pairs out of r vertices in a graph, then there is a cycle in the graph. Hence, since F is a feedback vertex set in G , there are at most $|M'| - 1$ connected components in B^3 . Recalling that $|M'| \leq 4k^2$, we obtain that $|B^3| \leq |M'| - 1 \leq 4k^2 - 1$.

Altogether, the number of connected components in $G - B$ is at most

$$\begin{aligned} 4k^3 + 2k^2 + \left| \bigcup_{\{x,y\} \subseteq B} B_{xy} \right| &\leq 4k^3 + 2k^2 + |B^1| + |B^2| + |B^3| \\ &\leq 4k^3 + 2k^2 + 2k^3 + k^2 - k + 9k^3 + k^2 + 4k^2 - 1 \\ &= 15k^3 + 8k^2 - k - 1. \end{aligned} \quad \square$$

Finally, putting all together, we can prove the the main result of this section.

Proof of Theorem 6.2. Let (G', k) be the input instance of SFVS. We compute the 2-core H of G . We apply Reduction Rules 6.4 to 6.8 exhaustively to obtain an equivalent instance (G, k) such that (G, k) is reduced. Next we apply Lemma 6.10 and obtain the set B in G with $|B| \leq 4k^2 + 2k$. Let \mathcal{C} denote the set of connected components in $G - B$. By Lemma 6.13, we know that $|\mathcal{C}| \leq 15k^3 + 8k^2 - k - 1$. Moreover, due

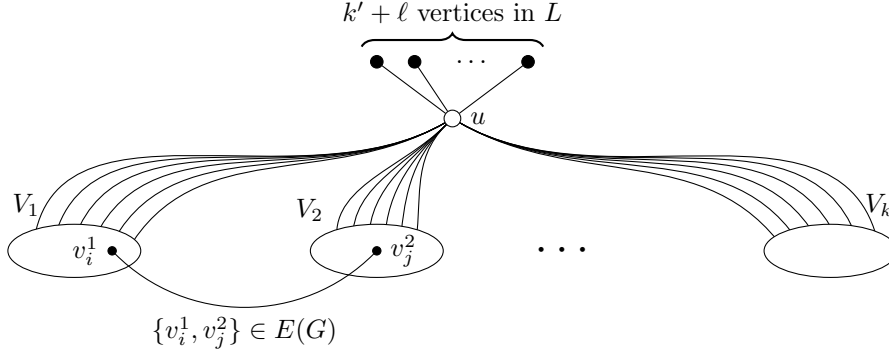


Figure 2: Sketch of the construction of graph G' on an input graph $G = (V = V_1 \cup \dots \cup V_k, E)$ as used in the proof of Theorem 6.14. The ellipses correspond to cliques with vertex sets V_i , $i \in \{1, \dots, k\}$.

to Lemma 6.12, for each $C \in \mathcal{C}$ it holds that $|V(C)| \leq (k+1) \cdot (12k+7)$. It follows that the number of vertices $|V(G)|$ in G is at most $|B| + |\mathcal{C}| \cdot \max_{C \in \mathcal{C}} |V(C)| \leq 4k^2 + 2k + (15k^3 + 8k^2 - k - 1) \cdot (k+1) \cdot (12k+7) \in O(k^5)$. \square

6.2. Small Secluded Feedback Vertex Set

In contrast to restricting the closed neighborhood of a feedback vertex set, restricting the open neighborhood by a parameter yields a W[1]-hard problem.

Problem (SMALL SECLUDED FEEDBACK VERTEX SET).

Input: A graph $G = (V, E)$ and two integers k, ℓ .

Question: Is there a set $S \subseteq V$ such that $G - S$ is cycle-free, $|S| \leq k$, and $|N_G(S)| \leq \ell$?

Theorem 6.14. SMALL SECLUDED FEEDBACK VERTEX SET is W[1]-hard with respect to ℓ .

Proof. We provide a parameterized reduction from MULTICOLORED INDEPENDENT SET (MIS): given a k -partite graph $G = (V, E)$ and its partite sets $V_1 \cup \dots \cup V_k = V$, the question is whether there is an independent set I of size k such that $I \cap V_i \neq \emptyset$ for each $i \in \{1, \dots, k\}$. MIS is W[1]-hard when parameterized by the size k of the independent set [20].

Let $G = (V, E)$ with partite sets $V_1 \cup V_2 \cup \dots \cup V_k = V$ be an instance of MIS. We can assume that for each $i \in \{1, \dots, k\}$ we have $|V_i| \geq 2$ and there is no edge $\{v, w\} \in E$ with $v, w \in V_i$. We create an instance (G', k', ℓ) of SMALL SECLUDED FEEDBACK VERTEX SET (SSFVS) with $k' = |V| - k$ and $\ell = k + 1$ as follows.

Construction: (Refer to Figure 2 for a sketch of the construction.) Initially, let $G' := G$. For each $i \in \{1, \dots, k\}$ turn V_i into a clique, that is, add the edge sets $\{\{a, b\} \mid a, b \in V_i, a \neq b\}$. Next, add to G' a vertex u and a set L of $k' + \ell$ vertices. Finally, connect each vertex in $V \cup L$ to u by an edge.

Correctness: We show that (G, k) is a yes-instance of MIS if and only if (G', k', ℓ) is a yes-instance of SSFVS.

(\Rightarrow) Let (G, k) be a yes-instance of MIS and let $I \subseteq V$ with $|I| = k$ be a multicolored independent set in G . We delete all vertices in $S := V(G') \setminus (I \cup L \cup \{u\})$ from G' . Observe that $|S| = |V| - k = k'$. Moreover, $N_{G'}(S) = k + 1 = \ell$. Since there is no edge between any two vertices in I , $G - S$ forms a star with center u and $k' + \ell + 1 + k$ vertices. Since every star is acyclic, (G', k', ℓ) is a yes-instance of SSFVS.

(\Leftarrow) Let (G', k', ℓ) be a yes-instance of SSFVS and let $S \subseteq V(G')$ be a solution. Observe that $G'[V_i \cup \{u\}]$ forms a clique of size $|V_i| + 1$ for each $i \in \{1, \dots, k\}$. Since the budget does not allow for deleting the vertex u (i.e. $u \notin S$), all but at most one vertex in each V_i must be deleted. Since $k' = |V| - k$ and $|V_i| \geq 2$ for all $i \in \{1, \dots, k\}$, S contains exactly $|V_i| - 1$ vertices of V_i for each $i \in \{1, \dots, k\}$. Hence, $|S| = |V| - k$ and $N_{G'}(S) = k + 1 = \ell$. Let $F := V \setminus S$ denote the set of vertices in V not contained in S . Recall that $|F| = k$ and $|F \cap V_i| = 1$ for all $i \in \{1, \dots, k\}$. Next, suppose there is an edge between two vertices $v, w \in F$. Since $u \notin S$ and u is incident to all vertices in V , the vertices u, v, w form a triangle in G' . This contradicts the

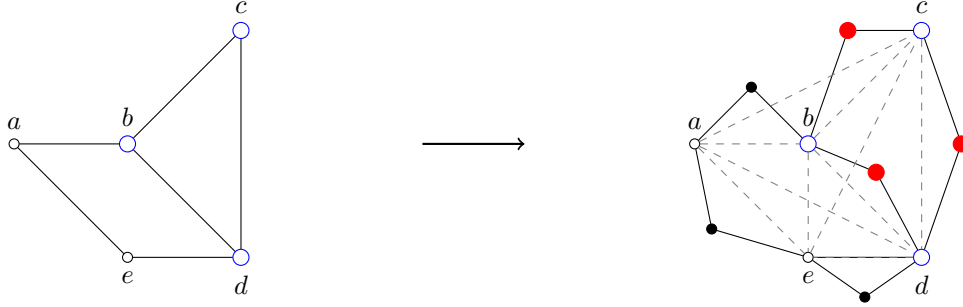


Figure 3: Example of the construction used in the proof of Theorem 7.1. The left-hand side shows the original graph, the right-hand side the graph constructed by the reduction, where the newly introduced edges between each pair of vertices of the original graph are drawn in dashed grey. The vertices introduced for each edge of the original graph are filled red and black, the corresponding new edges are drawn in black. Note that the enlarged, blue vertices of the original graph form a clique and that the vertices corresponding to the edges of said clique (enlarged, filled red) form an independent set in the new graph.

fact that S is a solution for (G', k', ℓ) , that is, that $G' - S$ is acyclic. It follows that $E(G'[F]) = \emptyset$, that is, no two vertices in F are connected by an edge. Together with $|F| = k$ and $|F \cap V_i| = 1$ for all $i \in \{1, \dots, k\}$, it follows that F forms a multicolored independent set in G . Thus, (G, k) is a yes-instance of MIS. \square

7. Independent Set

For INDEPENDENT SET, it makes little sense to bound the size of the closed neighborhood from above, as in this case the empty set always constitutes a solution. One might ask for an independent set with closed neighborhood as large as possible. However, for any inclusion-wise maximal independent set S , one has $N[S] = V$. Hence, this question is also trivial. Therefore, in this section we only consider the following problem.

Problem (LARGE SECLUDED INDEPENDENT SET (LSIS)).

Input: A graph $G = (V, E)$ and two integers k, ℓ .

Question: Is there an independent set $S \subseteq V$ such that $|S| \geq k$ and $|N_G(S)| \leq \ell$?

The case $\ell = |V|$ equals INDEPENDENT SET and, thus, LSIS is $W[1]$ -hard with respect to k . We show that LSIS is also $W[1]$ -hard when parameterized by $k + \ell$.

Theorem 7.1. LARGE SECLUDED INDEPENDENT SET is $W[1]$ -hard with respect to $k + \ell$.

We remark that the proof of Theorem 7.1 is identical to the $W[1]$ -hardness proof for CUTTING ℓ VERTICES [37]. However, for the sake of completeness, we present the proof in the remainder of this section.

Proof. We provide a polynomial-parameter transformation from CLIQUE parameterized by the solution size k .

Construction. Let (G, k) be an instance of CLIQUE and assume without loss of generality that $k < |V(G)| - 1$ (otherwise, solve the instance in polynomial time). We construct an equivalent instance (G', k', ℓ') of LARGE SECLUDED INDEPENDENT SET as follows (see Figure 3 for an example). Initially, let G' be an empty graph. Add all vertices of G to G' . Denote the vertex set by V . If two vertices of G are adjacent, we add a vertex to G' , that is, G' additionally to V contains the vertex set $X := \{x_{uv} \mid \{u, v\} \in E\}$. Next, connect x_{uv} to u and v , that is, add the edge set $E' = \{\{u, x_{uv}\}, \{v, x_{uv}\} \mid \{u, v\} \in E\}$. Finally, connect any two vertices in V by an edge. Graph G' consists of the vertex set $V \cup X$ and of the edge set $E' \cup \binom{V}{2}$. Observe that X forms an independent set in G' . Set $k' := \binom{k}{2}$ and $\ell' := k$. We claim that (G, k) is a yes-instance of CLIQUE if and only if (G', k', ℓ') is a yes-instance of LARGE SECLUDED INDEPENDENT SET.

(\Rightarrow) Let $C \subseteq V(G)$ be a clique of size $k = |C|$ in G . We claim that $X' := \{x_{u,v} \mid u, v \in C\}$ forms an independent set of size $\binom{k}{2}$ with $|N(X')| = k = \ell'$ in G' . Since $X' \subseteq X$, X' forms an independent set.

Moreover, since C is a clique of size k , there are $\binom{k}{2}$ edges in $G[C]$, and thus $|X'| = \binom{k}{2}$. By construction, each vertex in X is only adjacent to vertices in C . Hence, $|N(X')| = |C| = k$. Therefore, X' witnesses that (G', k', ℓ') is a yes-instance of LARGE SECLUDED INDEPENDENT SET.

(\Leftarrow) Let $U \subseteq V(G')$ form an independent set of size k' with open neighborhood of size upper-bounded by ℓ' . Suppose that $v \in V$ is contained in U (observe that U contains at most one vertex of V , as otherwise it would not be independent). Then $|N(U)| \geq |V| - 1 > k = \ell'$, which contradicts the choice of U . It follows that $U \cap V = \emptyset$, and hence $U \subseteq X$. By construction, for each $x_{uv} \in U$, the vertices u, v are contained in $N(U)$. Since each vertex in U corresponds to an edge in G , we have $\binom{k}{2}$ edges incident with at most k vertices. The only graph that fulfills this property is the complete graph on k vertices. Hence, G contains a clique of size k , and thus (G, k) is a yes-instance of CLIQUE(k). \square

8. Summary and Future Work

In this paper, we studied the problem of finding solutions with small neighborhood to classical combinatorial optimization problems in graphs. We presented computational complexity results for secluded and small secluded variants of s - t -SEPARATOR, q -DOMINATING SET, FEEDBACK VERTEX SET, \mathcal{F} -FREE VERTEX DELETION, and for the large secluded variant of INDEPENDENT SET. In the case of q -DOMINATING SET, we leave as an open question the parameterized complexity of SMALL p -SECLUDED q -DOMINATING SET, with $2p > q$, when parameterized by ℓ . Concerning SECLUDED \mathcal{F} -FREE VERTEX DELETION, we would like to point out that it is an interesting question which families \mathcal{F} exactly yield NP-hardness as opposed to polynomial-time solvability.

A natural way to generalize our results would be to consider vertex-weighted graphs and directed graphs. This generalization was already investigated by Chechik et al. [12] for SECLUDED PATH and SECLUDED STEINER TREE. Furthermore, replacing the bound on the open neighborhood in the case of small secludedness by a bound on the outgoing edges of a solution would be an interesting modification of the problem. The variation follows the idea of the concept of *isolation* [28, 30, 31, 34]. As the number of outgoing edges is at least as large as the open neighborhood, this might offer new possibilities for fixed-parameter algorithms. Finally, we focused on solutions of size at most or at least an integer k and did not discuss the case of size *exactly* k so far.

Acknowledgment. This research was initiated at the annual research retreat of the algorithms and complexity group of TU Berlin, held in Krölpa, Thuringia, Germany, from April 3rd till April 9th, 2016.

We would like to thank the anonymous referees of IPEC for comments that helped to improve the paper and for pointing us to the work of Fomin et al. [24]. The second author thanks Nikolay Karpov (St. Petersburg Department of the Steklov Institute of Mathematics of the Russian Academy of Sciences) for discussion on secluded problems.

Bibliography

- [1] V. Bafna, P. Berman, and T. Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 12(3):289–297, 1999. doi:[10.1137/S0895480196305124](https://doi.org/10.1137/S0895480196305124). Cited on page 24.
- [2] R. van Bevern. *Fixed-Parameter Linear-Time Algorithms for NP-hard Graph and Hypergraph Problems Arising in Industrial Applications*, volume 1 of *Foundations of Computing*. Universitätsverlag der TU Berlin, 2014. doi:[10.14279/depositonce-4131](https://doi.org/10.14279/depositonce-4131). Cited on page 19.
- [3] R. van Bevern. Towards optimal and expressive kernelization for d -Hitting Set. *Algorithmica*, 70(1):129–147, 2014. doi:[10.1007/s00453-013-9774-3](https://doi.org/10.1007/s00453-013-9774-3). Cited on pages 19 and 20.
- [4] R. van Bevern, H. Moser, and R. Niedermeier. Approximation and tidying—a problem kernel for s -Plex Cluster Vertex Deletion. *Algorithmica*, 62(3-4):930–950, 2012. ISSN 0178-4617. doi:[10.1007/s00453-011-9492-7](https://doi.org/10.1007/s00453-011-9492-7). Cited on page 19.
- [5] R. van Bevern, T. Fluschnik, G. B. Mertzios, H. Molter, M. Sorge, and O. Suchý. Finding secluded places of special interest in graphs. In *Proc. 11th IPEC*, volume 63 of *LIPICs*, pages 5:1–5:16. Schloss Dagstuhl, 2017. doi:[10.4230/LIPICs.IPEC.2016.5](https://doi.org/10.4230/LIPICs.IPEC.2016.5). Cited on page 1.
- [6] H. L. Bodlaender, P. G. Drange, M. S. Dregi, F. V. Fomin, D. Lokshantov, and M. Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45:317–378, 2016. doi:[10.1137/130947374](https://doi.org/10.1137/130947374). Cited on page 12.

- [7] M. Bruglieri, M. Ehr Gott, H. W. Hamacher, and F. Maffioli. An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics*, 154(9):1344–1357, 2006. doi:[10.1016/j.dam.2005.05.036](https://doi.org/10.1016/j.dam.2005.05.036). Cited on page 4.
- [8] T. N. Bui and C. Jones. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, 1992. doi:[10.1016/0020-0190\(92\)90140-Q](https://doi.org/10.1016/0020-0190(92)90140-Q). Cited on page 4.
- [9] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996. doi:[10.1016/0020-0190\(96\)00050-6](https://doi.org/10.1016/0020-0190(96)00050-6). Cited on pages 19 and 21.
- [10] L. Cai. Parameterized Complexity of Cardinality Constrained Optimization Problems. *The Computer Journal*, 51(1):102–121, 2008. doi:[10.1093/comjnl/bxm086](https://doi.org/10.1093/comjnl/bxm086). Cited on page 4.
- [11] L. Cai, S. M. Chan, and S. O. Chan. Random Separation: A New Method for Solving Fixed-Cardinality Optimization Problems. In *Proc. 2nd IWPEC*, pages 239–250. Springer, 2006. doi:[10.1007/11847250_22](https://doi.org/10.1007/11847250_22). Cited on pages 4 and 9.
- [12] S. Chechik, M. P. Johnson, M. Parter, and D. Peleg. Secluded connectivity problems. *Algorithmica*, 2016. doi:[10.1007/s00453-016-0222-z](https://doi.org/10.1007/s00453-016-0222-z). In press. Cited on pages 1, 2, 3, and 29.
- [13] B. Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi:[10.1016/0890-5401\(90\)90043-H](https://doi.org/10.1016/0890-5401(90)90043-H). Cited on page 8.
- [14] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshantov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:[10.1007/978-3-319-21275-3](https://doi.org/10.1007/978-3-319-21275-3). Cited on pages 5 and 6.
- [15] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 4th edition, 2010. Cited on page 5.
- [16] M. Dom, D. Lokshantov, and S. Saurabh. Kernelization lower bounds through colors and IDs. *ACM Transactions on Algorithms*, 11(2):13, 2014. doi:[10.1145/2650261](https://doi.org/10.1145/2650261). Cited on page 13.
- [17] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto, and F. Rosamond. Cutting up is hard to do: the parameterized complexity of k -Cut and related problems. *Electronic Notes in Theoretical Computer Science*, 78:209–222, 2003. doi:[10.1016/S1571-0661\(04\)81014-4](https://doi.org/10.1016/S1571-0661(04)81014-4). Cited on page 4.
- [18] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:[10.1007/978-1-4471-5559-1](https://doi.org/10.1007/978-1-4471-5559-1). Cited on page 5.
- [19] S. Fafianie and S. Kratsch. A shortcut to (sun)flowers: Kernels in logarithmic space or linear time. In *Proc. 40th MFCS*, volume 9235 of *LNCS*, pages 299–310. Springer, 2015. doi:[10.1007/978-3-662-48054-0_25](https://doi.org/10.1007/978-3-662-48054-0_25). Cited on pages 19 and 20.
- [20] M. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009. doi:[10.1016/j.tcs.2008.09.065](https://doi.org/10.1016/j.tcs.2008.09.065). Cited on page 27.
- [21] M. Fellows, J. Guo, H. Moser, and R. Niedermeier. A complexity dichotomy for finding disjoint solutions of vertex deletion problems. *ACM Transactions on Computation Theory*, 2(2):5, 2011. doi:[10.1145/1944857.1944860](https://doi.org/10.1145/1944857.1944860). Cited on page 19.
- [22] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006. doi:[10.1007/3-540-29953-X](https://doi.org/10.1007/3-540-29953-X). Cited on page 5.
- [23] F. Fomin, D. Lokshantov, N. Misra, and S. Saurabh. Planar F-Deletion: Approximation, kernelization and optimal FPT algorithms. In *Proc. 56th FOCS*, pages 470–479. IEEE Computer Society, 2012. doi:[10.1109/FOCS.2012.62](https://doi.org/10.1109/FOCS.2012.62). Cited on page 24.
- [24] F. Fomin, P. Golovach, and J. Korhonen. On the parameterized complexity of cutting a few vertices from a graph. In *Proc. 38th MFCS*, volume 8087 of *LNCS*, pages 421–432. Springer, 2013. doi:[10.1007/978-3-642-40313-2_38](https://doi.org/10.1007/978-3-642-40313-2_38). Cited on pages 3, 4, 8, and 29.
- [25] F. V. Fomin, P. A. Golovach, N. Karpov, and A. S. Kulikov. Parameterized complexity of secluded connectivity problems. *Theory of Computing Systems*, 61(3):795–819, 2017. doi:[10.1007/s00224-016-9717-x](https://doi.org/10.1007/s00224-016-9717-x). Cited on pages 2 and 3.
- [26] M. Gaertler. Clustering. In U. Brandes and T. Erlebach, editors, *Network Analysis: Methodological Foundations [outcome of a Dagstuhl seminar, 13-16 April 2004]*, volume 3418 of *LNCS*, pages 178–215. Springer, 2004. ISBN 3-540-24979-6. doi:[10.1007/978-3-540-31955-9_8](https://doi.org/10.1007/978-3-540-31955-9_8). Cited on page 2.
- [27] A. C. Giannopoulou, D. Lokshantov, S. Saurabh, and O. Suchý. Tree deletion set has a polynomial kernel (but no $\text{OPT}^{O(1)}$ approximation). *SIAM Journal on Discrete Mathematics*, 30(3):1371–1384, 2016. doi:[10.1137/15M1038876](https://doi.org/10.1137/15M1038876). Cited on pages 22 and 24.
- [28] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Isolation concepts for clique enumeration: Comparison and computational experiments. *Theoretical Computer Science*, 410(52):5384–5397, 2009. doi:[10.1016/j.tcs.2009.05.008](https://doi.org/10.1016/j.tcs.2009.05.008). Cited on pages 2, 4, and 29.
- [29] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory of Computing Systems*, 47(1):196–217, 2010. doi:[10.1007/s00224-008-9150-x](https://doi.org/10.1007/s00224-008-9150-x). Cited on page 19.
- [30] F. Hüffner, C. Komusiewicz, and M. Sorge. Finding highly connected subgraphs. In *Proc. 41st SOFSEM*, volume 8939 of *LNCS*, pages 254–265. Springer, 2015. doi:[10.1007/978-3-662-46078-8_21](https://doi.org/10.1007/978-3-662-46078-8_21). Cited on pages 2, 4, and 29.
- [31] H. Ito, K. Iwama, and T. Osumi. Linear-time enumeration of isolated cliques. In *Proc. 13th ESA*, volume 3669 of *LNCS*, pages 119–130. Springer, 2005. doi:[10.1007/11561071_13](https://doi.org/10.1007/11561071_13). Cited on pages 2, 4, and 29.
- [32] J. M. Kleinberg and É. Tardos. *Algorithm design*. Addison-Wesley, 2006. Cited on page 7.
- [33] C. Komusiewicz and M. Sorge. An algorithmic framework for fixed-cardinality optimization in sparse graphs applied to dense subgraph problems. *Discrete Applied Mathematics*, 193:145–161, 2015. doi:[10.1016/j.dam.2015.04.029](https://doi.org/10.1016/j.dam.2015.04.029). Cited on page 4.
- [34] C. Komusiewicz, F. Hüffner, H. Moser, and R. Niedermeier. Isolation concepts for efficiently enumerating dense subgraphs. *Theoretical Computer Science*, 410(38-40):3640–3654, 2009. doi:[10.1016/j.tcs.2009.04.021](https://doi.org/10.1016/j.tcs.2009.04.021). Cited on pages 2, 4, and 29.
- [35] J. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:[10.1016/0022-0000\(80\)90060-4](https://doi.org/10.1016/0022-0000(80)90060-4). Cited on page 19.
- [36] D. Lokshantov, N. Misra, G. Philip, M. Ramanujan, and S. Saurabh. Hardness of r -dominating set on graphs of diameter $(r + 1)$. In *Proc. 8th IPEC*, volume 8246 of *LNCS*, pages 255–267. Springer, 2013. doi:[10.1007/978-3-319-03898-8_22](https://doi.org/10.1007/978-3-319-03898-8_22).

Cited on page 13.

- [37] D. Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006. doi:[10.1016/j.tcs.2005.10.007](https://doi.org/10.1016/j.tcs.2005.10.007). Cited on pages 3, 4, and 28.
- [38] D. Marx, B. O’Sullivan, and I. Razgon. Finding Small Separators in Linear Time via Treewidth Reduction. *ACM Transactions on Algorithms*, 9(4):30:1–30:35, 2013. doi:[10.1145/2500119](https://doi.org/10.1145/2500119). Cited on page 9.
- [39] M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 182–191. IEEE Computer Society, 1995. doi:[10.1109/SFCS.1995.492475](https://doi.org/10.1109/SFCS.1995.492475). Cited on page 9.
- [40] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006. doi:[10.1093/acprof:oso/9780198566076.001.0001](https://doi.org/10.1093/acprof:oso/9780198566076.001.0001). Cited on page 5.
- [41] S. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269–287, 1983. doi:[10.1016/0378-8733\(83\)90028-X](https://doi.org/10.1016/0378-8733(83)90028-X). Cited on page 22.
- [42] S. Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010. doi:[10.1145/1721837.1721848](https://doi.org/10.1145/1721837.1721848). Cited on page 24.
- [43] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, 2000. Cited on page 5.