On the Parameterized Complexity of Reconfiguration Problems

Amer E. Mouawad¹ · Naomi Nishimura² · Venkatesh Raman³ · Narges Simjour⁴ · Akira Suzuki⁵,⁶

Received: 16 June 2015 / Accepted: 27 April 2016 / Published online: 10 May 2016
© Springer Science+Business Media New York 2016

Abstract We present the first results on the parameterized complexity of reconfiguration problems, where a reconfiguration variant of an optimization problem Q takes as input two feasible solutions S and T and determines if there is a sequence of reconfiguration steps, i.e. a reconfiguration sequence, that can be applied to transform S into T such that each step results in a feasible solution to Q. For most of the results in this paper, S and T are sets of vertices of a given graph and a reconfiguration step adds or removes a vertex. Our study is motivated by results establishing that for many NP-hard problems, the classical complexity of reconfiguration is PSPACE-complete. We address the question for several important graph properties under two natural parameterizations: k, a bound on the size of solutions, and ℓ, a bound on the length of reconfiguration sequences. Our first general result is an algorithmic paradigm, the reconfiguration kernel, used to obtain fixed-parameter tractable algorithms for recon-
configuration variants of Vertex Cover and, more generally, Bounded Hitting Set and Feedback Vertex Set, all parameterized by $k$. In contrast, we show that reconfiguring Unbounded Hitting Set is $\text{W}[2]$-hard when parameterized by $k + \ell$. We also demonstrate the $\text{W}[1]$-hardness of reconfiguration variants of a large class of maximization problems parameterized by $k + \ell$, and of their corresponding deletion problems parameterized by $\ell$; in doing so, we show that there exist problems in $\text{FPT}$ when parameterized by $k$, but whose reconfiguration variants are $\text{W}[1]$-hard when parameterized by $k + \ell$.

Keywords Reconfiguration · Parameterized complexity · Solution space · Vertex cover · Hitting set

1 Introduction

A reconfiguration variant of an optimization problem asks whether it is possible to transform a source feasible solution $S$ into a target feasible solution $T$ by a sequence of small incremental changes, i.e. a reconfiguration sequence, such that every intermediate solution is also feasible; other variants include finding a minimum-length reconfiguration sequence (if it exists). Reconfiguration problems model real-life dynamic situations in which we seek to transform a solution into a more desirable one, maintaining feasibility during the process. Moreover, the study of reconfiguration yields insights into the structure of the solution space of the underlying optimization problem, crucial for the design of efficient algorithms. We refer the reader to the survey by van den Heuvel [41] for a detailed overview of reconfiguration problems and their applications.

Motivated by real world situations as well as by trying to understand the structure of solution spaces, there has been a lot of interest in studying the complexity of reconfiguration problems. Problems for which reconfiguration has been studied include Vertex Colouring [3,5–9,27], List Edge-Colouring [24], Vertex Cover [35,36], Independent Set [20,23,25,26], Clique, Set Cover, Matching, Matroid Bases [23], Satisfiability [16,34], Shortest Path [4,28], and Dominating Set [18,19,39]. Most work has been limited to the problem of determining the existence of a reconfiguration sequence between two given solutions; for many $\text{NP}$-complete problems, this problem has been shown to be $\text{PSPACE}$-complete [23].

As there are typically exponentially many feasible solutions, the length of a reconfiguration sequence can be exponential in the size of the input instance. It is thus natural to ask whether reconfiguration problems become tractable if we allow the running time to depend on the length of the sequence; this approach suggests the use of the paradigm of parameterized complexity [13]. In this work, we explore reconfiguration in the framework of parameterized complexity under two natural parameterizations: $k$, a bound on the size of feasible solutions, and $\ell$, a bound on the length of reconfiguration sequences. One of our key results is that for most problems, the reconfiguration versions remain intractable in the parameterized framework when we parameterize by $\ell$. It is important to note that when $k$ is not bounded, the reconfiguration variants we study become easy (Sect. 2).
We present fixed-parameter algorithms for problems parameterized by \( k \) by nontrivial modifications to known parameterized algorithms for the problems. The paradigms of bounded search tree and kernelization typically work by exploring minimal solutions. However, a reconfiguration sequence may necessarily include non-minimal solutions. Any kernel that removes solutions (non-minimal or otherwise) may render finding a reconfiguration sequence impossible, as the missing solutions might appear in every reconfiguration sequence; we must thus ensure that the kernelization rules applied retain enough information to allow us to determine whether a reconfiguration sequence exists. To handle these difficulties, we introduce a general approach for parameterized reconfiguration problems. We use a reconfiguration kernel, showing how to adapt known kernels \([2,40]\) for Feedback Vertex Set (FVS), and a special kernel by Damaschke and Molokov \([11]\) for Bounded Hitting Set (BHS) (where the cardinality of each input set is bounded) to obtain polynomial reconfiguration kernels, with respect to \( k \). These results can be considered as interesting applications of kernelization, and a general approach for other similar reconfiguration problems.

As a counterpart to our result for Bounded Hitting Set, we show that reconfiguring Unbounded Hitting Set (UHS) or Dominating Set (DS) is \( \text{W}[2] \)-hard parameterized by \( k + \ell \). Finally, we show a general result on reconfiguration problems of hereditary properties and their parametric duals (see Sect. 2 for definitions), implying the \( \text{W}[1] \)-hardness of reconfiguring Independent Set (IS), Induced Forest (IF), and Induced Bipartite Subgraph (IBS) parameterized by \( k + \ell \) and Vertex Cover (VC), Feedback Vertex Set (FVS), and Odd Cycle Transversal (OCT) parameterized by \( \ell \).

2 Preliminaries

For general graph theoretic definitions we refer the reader to the book of Diestel \([12]\). Unless otherwise stated, we assume that each graph \( G \) is a simple, undirected graph with vertex set \( V(G) \) and edge set \( E(G) \), where \( |V(G)| = n \) and \( |E(G)| = m \). The open neighborhood, or simply neighborhood, of a vertex \( v \) is denoted by \( N_G(v) = \{u \mid (u,v) \in E(G)\} \), the closed neighborhood by \( N_G[v] = N_G(v) \cup \{v\} \). Similarly, for a set of vertices \( S \subseteq V(G) \), we define \( N_G(S) = \{v \mid (u,v) \in E(G), u \in S, v \notin S \} \) and \( N_G[S] = N_G(S) \cup S \). The degree of a vertex is \( |N_G(v)| \). We drop the subscript \( G \) when clear from context. A subgraph of \( G \) is a graph \( G' \) such that \( V(G') \subseteq V(G) \) and \( E(G') \subseteq E(G) \). The induced subgraph of \( G \) with respect to \( S \subseteq V(G) \) is denoted by \( G[S] ; G[S] \) has vertex set \( S \) and edge set \( E(G[S]) = \{(u,v) \mid u, v \in S, (u,v) \in E(G)\} \). We denote by \( \Delta(G) \) and \( \delta(G) \) the maximum degree and minimum degree of \( G \), respectively.

A walk of length \( \ell \) from \( v_0 \) to \( v_\ell \) in \( G \) is a vertex sequence \( v_0, \ldots, v_\ell \), such that for all \( i \in \{0, \ldots, \ell-1\} \), \((v_i, v_{i+1}) \in E(G)\). It is a path if all vertices are distinct. A path from vertex \( u \) to vertex \( v \) is also called a \( uv \)-path. The distance between two vertices \( u \) and \( v \) of \( G \), \( \text{dist}_G(u,v) \), is the length of a shortest \( uv \)-path in \( G \) (positive infinity if no such path exists).

The following definitions are based on optimization problems, each consisting of a polynomial-time recognizable set of valid instances, a set of feasible solutions for
each instance, and an objective function assigning a nonnegative integer value to each feasible solution. We sometimes use the modified big-Oh notation $O^*$ that suppresses all polynomially-bounded factors.

**Definition 1** The reconfiguration graph $R_Q(\mathcal{I}, A, k)$, consists of a node for each feasible solution to instance $\mathcal{I}$ of optimization problem $Q$, where the size of each solution is at least $k$ for $Q$ a maximization problem (of size at most $k$ for $Q$ a minimization problem, respectively), for positive integer $k$, and an edge between each pair of nodes corresponding to solutions in the binary adjacency relation $A$ on feasible solutions.

To avoid confusion, we refer to *vertices* in an input graph $G$ using lower case letters (e.g. $u, v$) and to the *nodes* in $R_Q(\mathcal{I}, A, k)$, and by extension their associated feasible solutions, using upper case letters (e.g. $S, T$). An edge in $R_Q(\mathcal{I}, A, k)$ corresponds to a *reconfiguration step*. A path in $R_Q(\mathcal{I}, A, k)$ is a sequence of such steps, i.e. a reconfiguration sequence. As an example, consider the case when $Q$ is the INDEPENDENT SET ($IS$) problem, i.e. given a graph $G$ and an integer $k$ determine whether $G$ has an independent set of size at least $k$. One natural definition of a binary adjacency relation $A$ on feasible solutions is to consider two independent sets to be adjacent whenever they are “one vertex away from each other”, i.e. one can be obtained from the other by the addition or the removal of a single vertex. With this definition, $R_{IS}(G, A, k)$ has a node for every independent set of $G$ of size at least $k$ and two nodes share an edge whenever the corresponding solutions are one vertex away from each other.

**Definition 2** For any problem $Q$, instance $\mathcal{I}$, adjacency relation $A$, positive integers $k$ and $\ell$, and feasible solutions $S_s$ and $S_t$, we define two reconfiguration problems:

- $Q$-REACH$(\mathcal{I}, A, S_s, S_t, k)$: For $S_s, S_t \in V(R_Q(\mathcal{I}, A, k))$, is there a path between $S_s$ and $S_t$ in $R_Q(\mathcal{I}, A, k)$?
- $Q$-BOUND$(\mathcal{I}, A, S_s, S_t, k, \ell)$: For $S_s, S_t \in V(R_Q(\mathcal{I}, A, k))$, is there a path of length at most $\ell$ between $S_s$ and $S_t$ in $R_Q(\mathcal{I}, A, k)$?

Going back to the INDEPENDENT SET example, in IS-REACH($G, A, S_s, S_t, k$), we are given a graph $G$, an integer $k$, and two independent sets $S_s$ and $S_t$ of size at most $k$, and the goal is to determine whether $S_s$ and $S_t$ belong to the same connected component in $R_{IS}(G, A, k)$. In the IS-BOUND($G, A, S_s, S_t, k, \ell$) problem, the goal is to determine whether there exists a path of length at most $\ell$ between $S_s$ and $S_t$ in $R_{IS}(G, A, k)$. We say $Q$-REACH is the reachability variant of $Q$, $Q$-BOUND is the bounded reachability variant of $Q$, and we refer to $Q$-REACH and $Q$-BOUND as reconfiguration problems for $Q$. The search variant of either problem asks for an actual reconfiguration sequence, i.e. the sequence of feasible solutions associated with a path in the reconfiguration graph. We drop the adjacency relation $A$ from the definition when clear from context.

Using the framework developed by Downey and Fellows [13], a parameterized reconfiguration problem includes in the input a parameter $p$ (typically $k$ or $\ell$). For a parameterized problem $Q$ with inputs of the form $(x, p)$, $|x| = n$ and $p$ a positive integer, $Q$ is fixed-parameter tractable (or in FPT) if it can be decided in $f(p)n^c$ time, where $f$ is an arbitrary computable function and $c$ is a constant independent of both $n$ and $p$. $Q$ is in the class XP if it can be decided in $n^f(p)$ time. $Q$ has a kernel of size $f(p)$ if there is an algorithm $A$ that transforms the input $(x, p)$ to $(x', p')$ such that $A$
runs in polynomial time (with respect to $|x|$ and $p$) and $(x, p)$ is a yes-instance if and only if $(x', p')$ is a yes-instance, $p' \leq g(p)$, $|x'| \leq f(p)$, and $f$ and $g$ are computable functions. Each problem in $\text{FPT}$ has a kernel [13], possibly of exponential (or worse) size.

We introduce the related notion of a reconfiguration kernel; it follows from the definition that a reconfiguration problem that has such a kernel is in $\text{FPT}$.

**Definition 3** A reconfiguration kernel of an instance of a parameterized reconfiguration problem $(x, p) = ((\mathcal{I}, A, S_i, S_t, k, \ell), p)$ is a set of $h(p)$ instances, for a computable function $h$, such that for $1 \leq i \leq h(p)$:

- for each instance in the set, $(x_i, p_i) = ((\mathcal{I}_i, A, S'_i, S''_i, k_i, \ell_i), p_i), \mathcal{I}_i, S'_i, S''_i, k_i, \ell_i$, and $p_i$ can all be computed in polynomial time,
- the size of each $x_i$ is bounded by $j(p)$, for a computable function $j$, and
- $(x, p)$ is a yes-instance if and only if at least one $(x_i, p_i)$ is a yes-instance.

The main hierarchy of parameterized complexity classes is $\text{FPT} \subseteq \text{W}[1] \subseteq \ldots \subseteq \text{XP}$, where $\text{W}$-hardness, shown using $\text{FPT}$-reductions, is the analogue of $\text{NP}$-hardness in classical complexity. A parameterized problem $Q$ $\text{FPT}$ reduces to a parameterized problem $Q'$ if there is an algorithm $A$ that transforms an instance $(\mathcal{I}, p)$ of $Q$ to an instance $(\mathcal{I}', p')$ of $Q'$ such that $A$ runs in time $f(p) \text{poly}(|\mathcal{I}|)$ where $f$ is a function of $p$, poly is a polynomial function, and $p' = g(p)$ for some computable function $g$. In addition, the transformation has the property that $(\mathcal{I}, p)$ is a yes-instance of $Q$ if and only if $(\mathcal{I}', p')$ is a yes-instance of $Q'$. It is known that standard parameterized versions (are there $p$ vertices that form the solution?) of $\text{CLIQUE}$ and $\text{INDEPENDENT SET}$ are complete for the class $\text{W}[1]$, and parameterized $\text{DOMINATING SET}$ is $\text{W}[2]$-complete. The reader is referred to the books by Flum, Grohe, and Niedermeier [15,37] for more on parameterized complexity.

Most problems we consider can be defined using graph properties, where a graph property $\Pi$ is a collection of graphs that is non-trivial if it is non-empty and does not contain all graphs. A graph property is polynomially decidable if for any graph $G$, it can be decided in polynomial time whether $G$ is in $\Pi$. The property $\Pi$ is hereditary if for any $G \in \Pi$, any induced subgraph of $G$ is also in $\Pi$. Examples of hereditary properties include graphs having no edges and graphs having no cycles. It is well known [31] that every hereditary property $\Pi$ has a forbidden set $\mathcal{F}_\Pi$, in that a graph has property $\Pi$ if and only if it does not contain any graph in $\mathcal{F}_\Pi$ as an induced subgraph.

For a graph property $\Pi$, we define two reconfiguration graphs, where solutions are subsets of the vertices of an input graph $G$ and two solutions are adjacent if they differ by the addition or removal of a single vertex. In other words, if $S$ and $T$ are two nodes in the reconfiguration graph then there exists an edge between $S$ and $T$ if and only if there exists a vertex $u \in V(G)$ such that $(S \setminus T) \cup (T \setminus S) = \{u\}$. Equivalently, for $S \Delta T = (S \setminus T) \cup (T \setminus S)$ the symmetric difference of $S$ and $T$, $S$ and $T$ share an edge in the reconfiguration graph if and only if $|S \Delta T| = 1$.

**Definition 4** The subset reconfiguration graph of $G$ with respect to $\Pi$, $R_\Pi^\Sigma(G, k)$, has a node for each $S \subseteq V(G)$ such that $|S| \geq k$ and $G[S]$ has property $\Pi$, and the
deletion reconfiguration graph of $G$ with respect to $\Pi$, $R^{\Pi}_{\text{DEL}}(G, k)$, has a node for each $S \subseteq V(G)$ such that $|S| \leq k$ and $G[V(G) \setminus S]$ has property $\Pi$.

Note that we can obtain $R^{\Pi}_{\text{DEL}}(G, |V(G)| - k)$ by replacing the set corresponding to each node in $R^{\Pi}_{\text{SUB}}(G, k)$ by its (setwise) complement. Any reconfiguration sequence from source feasible solution $S_s$ to target feasible solution $S_t$ in $R^{\Pi}_{\text{DEL}}(G, k)$ ($R^{\Pi}_{\text{SUB}}(G, k)$), which we sometimes denote by $\sigma = \langle S_0, S_1, \ldots, S_\ell \rangle$, for some $\ell$, has the following properties:

- $S_0 = S_s$ and $S_\ell = S_t$,
- $S_i$ is a feasible solution for all $0 \leq i \leq \ell$,
- $|S_i \Delta S_{i+1}| = 1$ for all $0 \leq i < \ell$, and
- $|S_i| \leq k$ (|$S_i| \geq k$) for all $0 \leq i \leq \ell$.

When dealing with underlying maximization problems, we say $k$ is the minimum allowed capacity; for minimization problems, $k$ is the maximum allowed capacity. We denote the length of $\sigma$ by $|\sigma|$. For $0 < i \leq \ell$, we say vertex $v \in V(G)$ is added at step $i$ if $v \notin S_{i-1}$ and $v \in S_i$. Similarly, a vertex $v$ is removed at step $i$ if $v \in S_{i-1}$ and $v \notin S_i$. A vertex $v \in V(G)$ is touched in the course of a reconfiguration sequence if $v$ is either added or removed at least once; it is untouched otherwise. A vertex is removable from (addable to) feasible solution $S$ if $S \setminus \{v\}$ ($S \cup \{v\}$) is also a feasible solution for $Q$. For any pair of consecutive solutions $(S_{i-1}, S_i)$ in $\sigma$, we say $S_i$ ($S_{i-1}$) is the successor (predecessor) of $S_{i-1}$ ($S_i$). A reconfiguration sequence $\sigma' = \langle S_0, S_1, \ldots, S_{\ell'} \rangle$ is a prefix of $\sigma = \langle S_0, S_1, \ldots, S_\ell \rangle$ if $\ell' < \ell$.

The following is a consequence of the fact that two nodes can differ by the removal or addition of a single vertex.

**Fact 1** The degree of each node in $R^{\Pi}_{\text{SUB}}(G, k)$ and each node in $R^{\Pi}_{\text{DEL}}(G, k)$ is at most $|V(G)|$.

**Definition 5** For any graph property $\Pi$, graph $G$, positive integer $k$, $S_s \subseteq V(G)$, and $S_t \subseteq V(G)$, we define the following decision problems:

- $\Pi$-DEL($G, k$): Is there $V' \subseteq V(G)$ such that $|V'| \leq k$ and $G[V(G) \setminus V'] \in \Pi$?
- $\Pi$-DEL-REACH($G, S_s, S_t, k$): For $S_s, S_t \in V(R^{\Pi}_{\text{DEL}}(G, k))$, is there a path between $S_s$ and $S_t$ in $R^{\Pi}_{\text{DEL}}(G, k)$?
- $\Pi$-DEL-BOUND($G, S_s, S_t, k, \ell$): For $S_s, S_t \in V(R^{\Pi}_{\text{DEL}}(G, k))$, is there a path of length at most $\ell$ between $S_s$ and $S_t$ in $R^{\Pi}_{\text{DEL}}(G, k)$?
- $\Pi$-SUB($G, k$): Is there $V' \subseteq V(G)$ such that $|V'| \geq k$ and $G[V'] \in \Pi$?
- $\Pi$-SUB-REACH($G, S_s, S_t, k$): For $S_s, S_t \in V(R^{\Pi}_{\text{SUB}}(G, k))$, is there a path between $S_s$ and $S_t$ in $R^{\Pi}_{\text{SUB}}(G, k)$?
- $\Pi$-SUB-BOUND($G, S_s, S_t, k, \ell$): For $S_s, S_t \in V(R^{\Pi}_{\text{SUB}}(G, k))$, is there a path of length at most $\ell$ between $S_s$ and $S_t$ in $R^{\Pi}_{\text{SUB}}(G, k)$?

We say that $\Pi$-DEL and $\Pi$-SUB are parametric duals of each other. In the $\Pi$-SUB problem, we seek a set of vertices of size at least $k$ inducing a subgraph in $\Pi$, whereas in $\Pi$-DEL, we seek a set of vertices of size at most $k$ whose complement
set induces a subgraph in $\Pi$. For example, for $\Pi$ the set of graphs with no edges, $\Pi$-DEL corresponds to the VERTEX COVER (VC) problem, $\Pi$-SUB corresponds to the INDEPENDENT SET problem, $\Pi$-DEL-REACH and $\Pi$-DEL-BOUND correspond to VC-REACH and VC-BOUND, respectively, while $\Pi$-SUB-REACH and $\Pi$-SUB-BOUND correspond to IS-REACH and IS-BOUND, respectively. As noted in the introduction, both the reachability and bounded reachability variants of $\Pi$-DEL and $\Pi$-SUB become polynomial-time solvable if we drop the constraint on the size of feasible solutions, i.e. the parameter $k$. We illustrate with the example of VERTEX COVER. Since any superset of a vertex cover of a graph $G$ is also a vertex cover of $G$, we can, starting from $S_0$, add all vertices of the graph to reach a vertex cover of size $n$ and then remove all vertices in $V(G) \setminus S_t$ to reach $S_t$.

3 Fixed-Parameter Tractability Results

We first observe that for any polynomial-time decidable graph property, $\Pi$-DEL-BOUND and $\Pi$-SUB-BOUND are in $\text{XP}$ when parameterized by $\ell$; we conduct breadth-first search on the reconfiguration graph starting at $S_0$, stopping either upon discovery of $S_t$ or upon completing the exploration of $\ell$ levels. Fact 1 implies a bound of at most $n^\ell$ vertices to explore in total.

Fact 2 For any polynomially-decidable graph property $\Pi$, $\Pi$-DEL-BOUND $\in \text{XP}$ and $\Pi$-SUB-BOUND $\in \text{XP}$ when parameterized by $\ell$.

The situations differ for $\Pi$-DEL-REACH and $\Pi$-SUB-REACH when parameterized by $k$, as the latter is based on a maximization problem while the former is based on a minimization problem. In other words, we can enumerate all feasible solutions of size at most $k$ in $O(n^k)$ time to solve $\Pi$-DEL-REACH (or $\Pi$-DEL-BOUND) parameterized by $k$ but the same “easy” argument does not hold for $\Pi$-SUB-REACH (or $\Pi$-SUB-BOUND) parameterized by $k$. Already, Fact 3 implies that, in some sense, dealing with reachability and bounded reachability variants of maximization problems might be “harder” in the parameterized setting.

Fact 3 For any polynomially-decidable graph property $\Pi$, $\Pi$-DEL-REACH $\in \text{XP}$ and $\Pi$-DEL-BOUND $\in \text{XP}$ when parameterized by $k$.

For a graph $G$ and $S_s, S_t \subseteq V(G)$, we partition $V(G)$ into the sets $C_{st} = S_s \cap S_t$ (vertices common to $S_s$ and $S_t$), $S_s \setminus t = S_s \setminus S_t$ (vertices to be removed from $S_s$ in the course of a reconfiguration sequence), $S_t \setminus s = S_t \setminus S_s$ (vertices to be added to form $S_t$), and $O_{st} = V(G) \setminus (S_s \cup S_t)$ (all other vertices). Furthermore, we can partition $C_{st}$ into two sets $C_F$ and $C_M = C_{st} \setminus C_F$, where a vertex is in $C_F$ if and only if it is in every feasible solution of size bounded by $k$.

The following fact is a consequence of the definitions above, the fact that $\Pi$ is hereditary, and the observations that $G[S_s \setminus t]$ and $G[O_{st}]$ are both subgraphs of $G[V(G) \setminus S_t]$, and $G[S_{s \setminus t}]$ and $G[O_{st}]$ are both subgraphs of $G[V(G) \setminus S_s]$.

Fact 4 For any hereditary property $\Pi$, graph $G$, and $S_s, S_t \subseteq V(G)$ such that $G[V(G) \setminus S_s]$, $G[V(G) \setminus S_t] \in \Pi$, the graphs $G[O_{st}]$, $G[S_{s \setminus t}]$, and $G[S_{t \setminus s}]$ are all in $\Pi$. 

 Springer
In any reconfiguration sequence, each vertex in \( S_{x_1} \) must be removed and each vertex in \( S_{x_2} \) must be added. In fact, since \( \ell \) implies a bound on the total number of vertices that can be touched in a reconfiguration sequence, setting \( \ell = |S_x \Delta S_t| = |S_{x_1}| + |S_{x_2}| \) drastically simplifies the bounded reachability problem.

**Observation 1** For any polynomially-decidable graph property \( \Pi \), if \( \ell = |S_{x_1}| + |S_{x_2}| \), then \( \Pi\text{-}\text{Del-Bound} \) and \( \Pi\text{-}\text{Sub-Bound} \) can be solved in \( O^*(2^\ell) \) time, and hence are fixed-parameter tractable when parameterized by \( \ell \).

**Proof** Since each vertex in \( S_{x_1} \) must be added and each vertex in \( S_{x_2} \) removed, in \( \ell \) steps we can touch each vertex in \( S_{x_1} \cup S_{x_2} \) exactly once; all vertices in \( V(G) \setminus (S_{x_1} \cup S_{x_2}) \) remain untouched.

Any node in the path between \( S_x \) and \( S_t \) in \( R^\Pi_{\text{Sub}}(G, k) \) represents a set \( C_{st} \cup B \), where \( B \) is a subset of \( S_{x_1} \cup S_{x_2} \). As \( |S_{x_1}| + |S_{x_2}| = \ell \), there are only \( 2^\ell \) choices for \( B \). Our problem then reduces to finding the shortest path between \( S_x \) and \( S_t \) in the subgraph of \( R^\Pi_{\text{Sub}}(G, k) \) induced on the \( 2^\ell \) relevant nodes; the bound follows from the fact that the number of edges is at most \( 2^\ell |V(G)| \), a consequence of Fact 1. The same argument holds for \( R^\Pi_{\text{Del}}(G, k) \).

In contrast, we show in the next section that for most hereditary properties, reconfiguration problems are hard when parameterized by \( \ell \).

We conclude this section with an observation which relates the parameterized complexity of \( \Pi\text{-}\text{Del-Bound} \) and \( \Pi\text{-}\text{Sub-Bound} \) when both are parameterized by \( \ell \); proving fixed-parameter tractability of either problem on a given graph class will be enough to imply fixed-parameter tractability of the other.

**Observation 2** Given \( \Pi \) and a collection of graphs \( \mathcal{C} \), \( \Pi\text{-}\text{Del-Bound} \) parameterized by \( \ell \) is fixed-parameter tractable on \( \mathcal{C} \) if and only if \( \Pi\text{-}\text{Sub-Bound} \) is.

**Proof** Given an instance \((G, S_x, S_t, k, \ell)\) of \( \Pi\text{-}\text{Sub-Bound} \), where \( G \in \mathcal{C} \), we solve the \( \Pi\text{-}\text{Del-Bound} \) instance \((G, V(G) \setminus S_x, V(G) \setminus S_t, n - k, \ell)\). Note that the parameter \( \ell \) remains unchanged.

For any node \( S \) in \( R^\Pi_{\text{Sub}}(G, k) \) there exists a corresponding node \( S' = V(G) \setminus S \) in \( R^\Pi_{\text{Del}}(G, n - k) \), i.e. if \( G[S] \) is in \( \Pi \) then \( G[V(G) \setminus S'] = G[S] \) is also in \( \Pi \). Hence, it is not hard to see that there exists a path between the nodes \( S_x \) and \( S_t \) in \( R^\Pi_{\text{Sub}}(G, k) \) if and only if there exists a path of the same length between the nodes \( V(G) \setminus S_x \) and \( V(G) \setminus S_t \) in \( R^\Pi_{\text{Del}}(G, n - k) \). The same argument holds for the other direction.

### 3.1 Bounded Hitting Set

Here, we prove the parameterized tractability of reachability and bounded reachability variants of certain superset-closed \( k \)-subset problems when parameterized by \( k \); a \( k \)-subset problem is a parameterized minimization problem \( Q \) whose solutions for an instance \((I, k)\) are all subsets of size at most \( k \) of a finite domain set \( D \). We say \( Q \) is superset-closed if any superset of size at most \( k \) of a solution of \( Q \) is also a solution of \( Q \). For example, VERTEX COVER is a superset-closed problem but INDEPENDENT SET is not.

\( \odot \) Springer
Theorem 1  If a $k$-subset problem $Q$ is superset-closed and has an $\textbf{FPT}$ algorithm to enumerate all its minimal solutions of cardinality at most $k$, the number of which is bounded by a function of $k$, then $Q$-REACH and $Q$-BOUND parameterized by $k$ are in $\textbf{FPT}$.

Proof  By enumerating all minimal solutions of an instance of $Q$ of cardinality at most $k$, we compute the set $M$ of all elements $v$ of the domain set such that $v$ is in a minimal solution. For $(I, S_r, S_t, k, \ell)$ an instance of $Q$-BOUND, we show that there exists a reconfiguration sequence from $S_r$ to $S_t$ if and only if there exists a reconfiguration sequence from $S_r \cap M$ to $S_t \cap M$ that touches only elements of $M$.

Each set $U$ in the reconfiguration sequence from $S_r$ to $S_t$ is a solution, and hence contains at least one minimal solution in $U \cap M$; $U \cap M$ is a superset of the minimal solution and hence also a solution. Moreover, since any two consecutive solutions $U$ and $U'$ in the sequence differ by a single element, $U \cap M$ and $U' \cap M$ differ by at most a single element. By replacing each subsequence of identical sets by a single set, we obtain a reconfiguration sequence from $S_r \cap M$ to $S_t \cap M$ that uses only subsets of $M$.

The reconfiguration sequence from $S_r \cap M$ to $S_t \cap M$ using only subsets of $M$ can be extended to a reconfiguration sequence from $S_r$ to $S_t$ by transforming $S_r$ to $S_r \cap M$ in $|S_r \setminus M|$ steps and transforming $S_t \cap M$ to $S_t$ in $|S_t \setminus M|$ steps. Note that this might not be a shortest reconfiguration sequence from $S_r$ to $S_t$. In this sequence, each vertex in $C_{st} \setminus M$ is removed from $S_r$ to form $S_r \setminus M$ and then readded to $S_t$ from $S_t \setminus M$. For each vertex $v \in C_{st} \setminus M$, we can choose instead to add $v$ to each solution in the sequence, thereby decreasing $\ell$ by two (the steps needed to remove and then readd $v$) at the cost of increasing by one the capacity used in the sequence from $S_r \cap M$ to $S_t \cap M$.

This choice can be made independently for each of these $E = |C_{st} \setminus M|$ vertices. Consequently, $(I, S_r, S_t, k, \ell)$ is a yes-instance for $Q$-BOUND if and only if one of the $E + 1$ reduced instances $(I, S_r \cap M, S_t \cap M, k - e, \ell - 2(E - e))$, for $0 \leq e \leq E$ and $E = |C_{st} \setminus M|$, is a yes-instance for $Q'$-BOUND, where we define $Q'$ as a $k$-subset problem whose solutions for an instance $(I, k)$ are solutions of instance $(I, k)$ of $Q$ that are contained in $M$.

To show that $Q'$-BOUND is in $\textbf{FPT}$, we observe that the number of nodes in the reconfiguration graph for $Q'$ is bounded by a function of $k$. Each solution of $Q'$ is a subset of $M$, yielding at most $2^{|M|}$ nodes, and $|M|$ is bounded by a function of $k$.  \hfill $\square$

As a consequence, the reachability and bounded reachability variants of (minimization) problems such as $\text{Bounded Hitting Set}$ (BHS), $\text{Feedback Arc Set}$ in $\text{Tournaments}$ (FAST), and $\text{Minimum Weight SAT}$ in $\text{Bounded Monotone CNF Formulas}$ (MIN- BMCNF- SAT) are proved to be in $\textbf{FPT}$ when parameterized by $k$. In the BHS problem, given a finite universe $\mathcal{U}$, a family of sets $\mathcal{F} \subseteq 2^\mathcal{U}$ each of size at most a constant $c$, and an integer $k$, the goal is to determine whether there exists a hitting set $H \subseteq \mathcal{U}$ such that $|H| \leq k$ and for every set $F \in \mathcal{F}$ we have $F \cap U \neq \emptyset$. A tournament is a directed graph $T$ such that for every pair of vertices $u, v \in V(T)$ exactly one of $(u, v)$ or $(v, u)$ is a directed edge (also often called an arc) of $T$. A set of arcs $A$ of a directed graph $G$ is called a feedback arc set if every directed cycle of $G$ contains an arc from $A$. In other words, the removal of $A$ from $G$ turns it into a directed acyclic graph. In the FAST problem, we are given a tournament $T$ and a
non-negative integer \( k \). The objective is to decide whether \( T \) has a feedback arc set of size at most \( k \). Finally, in the \textsc{MIN- BMCNF- SAT} problem, we are given a Boolean formula in conjunctive normal form on \( n \) variables and \( m \) clauses and an integer \( k \). Each clause consists of at most \( c \) positive literals, where \( c \) is some fixed constant. A solution to the \textsc{MIN- BMCNF- SAT} problem is a set of at most \( k \) variables that are set to true in a satisfying assignment.

**Corollary 1** \textsc{BHS- Reach}, \textsc{BHS- Bound}, \textsc{FAST- Reach}, \textsc{FAST- Bound}, \textsc{MIN- BMCNF- SAT- Reach}, and \textsc{MIN- BMCNF- SAT- Bound} parameterized by \( k \) are in \textsc{FPT}.

**Proof** The \textsc{BHS}, \textsc{FAST}, and \textsc{MIN- BMCNF- SAT} problems are all superset-closed. For each we demonstrate the applicability of Theorem 1 by giving an \textsc{FPT} algorithm that uses standard techniques to enumerate all minimal solutions, the number of which is bounded by a function of \( k \).

We can devise a search tree algorithm that gradually constructs minimal hitting sets of instances of \textsc{Bounded Hitting Set}, producing all minimal hitting sets of size at most \( k \) in its leaves. Consider an instance of \textsc{Bounded Hitting Set}, where the cardinality of each set is bounded by a constant \( c \). At each non-leaf node, the algorithm chooses a set that has not yet been hit, and branches on all possible ways of hitting this set, including one of the (at most \( c \)) elements in the set in each branch. Since we are not interested in hitting sets of cardinality more than \( k \), we do not need to search beyond depth \( k \) in the tree, proving an upper bound of \( c^k \) on the number of leaves, and an upper bound of \( O^*(c^k) \) on the enumeration time.

For the \textsc{Feedback Arc Set in Tournaments} problem, a tournament is acyclic if and only if it does not contain a directed cycle of length three \([1]\), and a set of arcs is a minimal feedback arc set in a tournament if and only if reversing its arcs in the tournament results in an acyclic tournament \([38]\). Therefore, at each non-leaf node in a search tree for this problem, there is always a cycle \( C \) of length three with which every feedback arc set shares at least one arc. The algorithm can thus solve the problem recursively by branching, choosing one of the three arcs of \( C \) to reverse. As in the previous algorithm, since we are not interested in feedback arc sets of cardinality more than \( k \), the search can be terminated at depth \( k \), proving an upper bound of \( 3^k \) on the number of minimal \( k \)-feedback arc sets in tournaments, and an upper bound of \( O^*(3^k) \) on the running time of this enumeration algorithm.

Finally, Misra et al. \([33]\) give a search tree algorithm for \textsc{MIN- BMCNF- SAT}, where every clause has at most \( c \) literals for some constant \( c \). At each node, the algorithm chooses a clause (whose literals are all positive), and branches on all possible ways of satisfying the clause, setting one variable to true in each branch. If there is no such clause, the formula is satisfied with no increase in the number of true variables, that is, by setting every non-assigned variable to false. As before, the algorithm stops the search when it reaches a depth of \( k \), proving an upper bound of \( c^k \) on the number of satisfying assignments, and an upper bound of \( O^*(c^k) \) on the enumeration time. \(\square\)

For \textsc{Bounded Hitting Set}, the proof of Theorem 1 can be strengthened to develop a polynomial reconfiguration kernel. In fact, we use the ideas in Theorem 1 to adapt a special kernel that retains all minimal \( k \)-hitting sets in the reduced instances \([11]\).
Theorem 2  BHS-Reach and BHS-Bound parameterized by $k$ each admit a polynomial reconfiguration kernel.

Proof We consider the formulation of the BHS problem using hypergraphs. That is, given an instance $(\mathcal{U}, \mathcal{F}, k)$ of BHS, we construct a hypergraph $G$ consisting of $|\mathcal{U}|$ vertices and a family of $|\mathcal{F}|$ subsets of vertices called hyperedges. Each hyperedge consists of at most $c$ vertices, for some constant $c$. Now, a $k$-hitting set is a set of at most $k$ vertices that intersects all hyperedges in $G$. We let $(G, S_y, S_t, k, \ell)$ be the resulting instance of BHS-Bound. Each of $S_y$ and $S_t$ is a hitting set of size at most $k$, that is, a set of vertices intersecting each hyperedge in $G$.

Damaschke and Molokov [11] showed that, given $G$, and assuming $G$ has a $k$-hitting set, we can construct a hypergraph $G'$ which satisfies the following properties:

- The vertex set of $G'$ is contained in that of $G$,
- $G'$ has exactly the same minimal $k$-hitting sets as $G$,
- $G'$ contains at most $k^c$ hyperedges and at most $(c - 1)k^c + k$ vertices, and
- we can compute $G'$ from $G$ in $O^*(k^{c-1})$ time.

We form a reconfiguration kernel using the polynomial-time algorithm of Damaschke and Molokov to obtain $G'$. $V(G')$ includes all minimal $k$-hitting sets, $V(G')$ is of size at most $(c - 1)k^c + k$, and the $k$-hitting sets for $G'$ are those $k$-hitting sets for $G$ that are subsets of $V(G')$. Therefore, as in the proof of Theorem 1, $(G, S_y, S_t, k, \ell)$ is a yes-instance for BHS-Bound if and only if one of the $E + 1$ reduced instances $(G', S_y \cap V(G'), S_t \cap V(G'), k - e, \ell - 2(E - e))$, for $0 \leq e \leq E$, is a yes-instance. Notice that unlike in the proof of Theorem 1, here the set containing all minimal solutions can be computed in polynomial time, whereas Theorem 1 guarantees only a fixed-parameter tractable procedure. \hfill \Box

Bounded Hitting Set generalizes any deletion (minimization) problem for a hereditary property with a finite forbidden set, which implies Corollary 2. In other words, we can reduce any instance $(G, k)$ of $\Pi$-Del, for $\Pi$ a hereditary property with a finite forbidden set $\mathcal{F}_\Pi$, to an instance $(\mathcal{U}, \mathcal{F}, k)$ of Bounded Hitting Set as follows. Since $\mathcal{F}_\Pi$ is a finite set, we know that there exists a constant $c$ such that any graph in $\mathcal{F}_\Pi$ contains at most $c$ vertices. Moreover, the size of $\mathcal{F}_\Pi$ is also bounded by some constant $c'$, and neither $c$ nor $c'$ depends on the input size. Therefore, we can enumerate all forbidden induced subgraphs in $G$ in $O(n^c)$ time. We construct an instance of Bounded Hitting Set by setting $\mathcal{U} = V(G)$ and adding one set (of size at most $c$) to $\mathcal{F} \subseteq 2^\mathcal{U}$ containing all vertices of each forbidden subgraph found during the enumeration phase. It is not hard to see that there is a one-to-one mapping between solutions of size at most $k$, i.e. a set $H \subseteq \mathcal{U}$ is a hitting set of size at most $k$, if and only if the corresponding vertices form a solution to the $\Pi$-Del instance. Hence, we can in a similar manner reduce both $\Pi$-Del-Reach and $\Pi$-Del-Bound to BHS-Reach and BHS-Bound, respectively.

Corollary 2 If $\Pi$ is a hereditary graph property with a finite forbidden set, then $\Pi$-Del-Reach and $\Pi$-Del-Bound parameterized by $k$ each admit a polynomial reconfiguration kernel.

© Springer
3.2 Undirected Feedback Vertex Set

Corollary 2 does not apply to Feedback Vertex Set, for which the associated hereditary graph property is the collection of all forests; the forbidden set is the set of all cycles and hence is not finite. Theorem 1 does not apply to Feedback Vertex Set either, since the number of minimal solutions exceeds $f(k)$ if the input graph includes a cycle of length $f(k) + 1$, for any function $f$. While it may be possible to adapt the compact enumeration of minimal feedback vertex sets [17] for reconfiguration, we develop a reconfiguration kernel by modifying specific kernels for the problem. We are given an undirected graph and two feedback vertex sets $S_s$ and $S_t$ of size at most $k$.

We make use of Bodlaender’s cubic kernel [2] and Thomassé’s quadratic kernel [40] for the Feedback Vertex Set problem, modifying reduction rules to allow the reconfiguration sequence to use non-minimal solutions.

Our presentation closely follows the one given in the recent book of Cygan et al. [10] and proofs are included for completeness. As reduction rules may introduce multiedges, forming a multigraph, we need some additional notation. For a vertex $v \in V(G)$, we use $d_G(v)$ to denote the number of edges incident on $v$ in the multigraph $G$. We also use the convention that a loop at a vertex $v$ contributes two to $d_G(v)$. Given a vertex $v \in V(G)$, a $v$-flower of order $k$ is a set of $k$ cycles in $G$ whose pairwise intersection is exactly $\{v\}$. A $q$-star, $q \geq 1$, is a graph with $q + 1$ vertices and $q$ edges, one vertex of degree $q$ and all other vertices of degree 1. For $G$ a bipartite graph with vertex bipartition $(A, B)$, a set of edges $M \subseteq E(G)$ is called a $q$-expansion of $A$ into $B$ if (i) every vertex of $A$ is incident with exactly $q$ edges of $M$ and (ii) $M$ saturates exactly $q|A|$ vertices in $B$.

The following Lemma [10] will guide our kernelization algorithm. In particular, our goal is to make sure that each vertex in $V(G)$ is incident to at most $O(k^2)$ edges and each vertex in $V(G) \setminus (S_s \cup S_t)$ is incident to at least three edges. Consequently, we obtain instances with $O(k^3)$ vertices and edges.

**Lemma 1** ([10]) Let $G$ be a (multi) graph and let $S_s$ and $S_t$ be two feedback vertex sets of $G$ of size at most $k$. If for all $v \in V(G)$ we have $d_G(v) \leq d$ and for all $v \in V(G) \setminus (S_s \cup S_t)$ we have $d_G(v) \geq 3$, then $|V(G)| \leq 2(d+1)k$ and $|E(G)| \leq 4dk$.

**Proof** We let $E_{cross} \subseteq E(G)$ denote the set of edges with one endpoint in $X = S_s \cup S_t$ and the other in $Y = V(G) \setminus X$. Since $d_G(v) \leq d$, for all $v \in V(G)$, $E_{cross} \leq d|X| \leq 2dk$. On the other hand, since $G[Y]$ is a forest, $G[Y]$ contains less than $|Y|$ edges and has average degree fewer than two. Therefore,

$$\sum_{v \in Y} d_G(v) = \sum_{v \in Y} d_G([v] \cup X)(v) + \sum_{v \in Y} d_G[Y](v) \leq |E_{cross}| + 2|Y|.$$  

Given that $d_G(v) \geq 3$, for all $v \in Y$, we get

$$\sum_{v \in Y} d_G(v) \geq 3|Y|.$$
Putting it all together, we obtain $|Y| \leq |E_{\text{cross}}| \leq d|X|$ and hence

$$|V(G)| = |X| + |Y| \leq (d + 1)|X| \leq 2(d + 1)k.$$ 

To bound the number of edges, we again use the fact that $d_G(v) \leq d$, for all $v \in V(G)$, i.e. there are at most $d|X|$ edges incident to $|X|$ and all other edges of $G$ are contained in $G[Y]$. As $G[Y]$ is a forest, it has fewer than $|Y|$ edges. Consequently,

$$E(G) \leq d|X| + |Y| \leq 2d|X| \leq 4dk.$$ 

This completes the proof.

In the following reduction rules, we take into account the roles of $C_{st}$, $S_{s/t}$, $S_{t/s}$, and $O_{st}$. Recall that we partition $C_{st}$ into two sets $C_F$ and $C_M = C_{st} \setminus C_F$, where a vertex is in $C_F$ if and only if it is in every feasible solution of size at most $k$. We introduce $E$ to denote how much capacity we can “free up” for use in a reduced instance by removing vertices and then reading them.

**Reduction Rule 1** If there exists $v \in V(G)$ such that $d_G(v) = 0$, delete $v$ from $G$. If $v$ is in $S_{s/t} \cup S_{t/s}$, subtract 1 from $\ell$. If $v$ is in $C_{st}$, increment $E$ by 1.

**Reduction Rule 2** If there exists $v \in V(G)$ such that $d_G(v) = 1$, delete $v$ and its incident edge from $G$. If $v$ is in $S_{s/t} \cup S_{t/s}$, subtract 1 from $\ell$. If $v$ is in $C_{st}$, increment $E$ by 1.

**Reduction Rule 3** If there are three or more edges $(v, w) \in E(G)$, delete all but two.

**Reduction Rule 4** If there exists $v \in V(G)$ such that $d_G(v) = 2$ and $v$ is in $O_{st}$, delete $v$ and its incident edges from $G$ and add an edge between its neighbours $w$ and $x$ (resulting in a self-loop if $w = x$).

**Reduction Rule 5** If $v \in V(G)$ has a self-loop, delete $v$ and all incident edges and decrease $k$ by 1.

Before we can state our remaining two reduction rules, which bound the maximum number of edges incident on each vertex, we need the following results.

**Lemma 2** ([10]) Given a (multi) graph $G$, an integer $k$, and a vertex $v \in V(G)$, there is a polynomial-time algorithm that either finds a $v$-flower of order $k + 1$ or finds a set $Z_v$ such that $Z_v \subseteq V(G)$ intersects all cycles passing through $v$, $|Z_v| \leq 2k$, and there are at most $2k$ edges incident to $v$ and with second endpoint in $Z_v$.

**Lemma 3** ([10,40]) Let $q$ be a positive integer and $G$ be a bipartite graph with bipartition $(A, B)$ such that $|B| \geq q|A|$ and there are no isolated vertices in $B$. Then, there exist nonempty vertex sets $X \subseteq A$ and $Y \subseteq B$ such that:

- $X$ has a $q$-expansion into $Y$ and
- no vertex in $Y$ has a neighbour outside $X$, i.e. $N_G(Y) \subseteq X$.

Furthermore, the sets $X$ and $Y$ can be found in time polynomial in the size of $G$.

For every vertex $v \in V(G)$ such that $d_G(v) \geq 2k(k + 2) + 5k$, we apply the algorithm of Lemma 2. If the algorithm finds a $v$-flower of order $k + 1$, the following reduction rule allows us to deal with it.
Reduction Rule 6 If there exists a $v$-flower of order $k + 1$ or more, delete $v$ and all incident edges and decrease $k$ by 1.

Hence, in what follows we assume that no such flower was found but instead we have a set $Z_v$ of size at most $2k$ such that $Z_v \subseteq V(G)$ intersects all cycles passing through $v$. Consider the connected components of the graph $G[V(G) \setminus (Z_v \cup \{v\})]$. At most $k$ of those components can contain a cycle, as otherwise we have $k + 1$ vertex disjoint cycles which contradicts the fact that $G$ has a feedback vertex set of size at most $k$. Moreover, for every component $D$ of $G[V(G) \setminus (Z_v \cup \{v\})]$, we have $|N_G(v) \cap V(D)| \leq 1$ (since $Z_v$ intersects all cycles passing through $v$). In other words, $v$ has at most one neighbor in any component and out of those components at most $k$ are not trees. Let $D = \{D_1, D_2, \ldots, D_p\}$ denote those trees in which $v$ has at least one neighbor and such that in each $D_i$, $1 \leq i \leq p$, there exists at least one leaf with a neighbor in $Z_v$. After exhaustive application of Reduction Rule 4, for all vertices $v \in O_{st}$ we have $d_G(v) \geq 3$ and hence at most $2k$ vertices (vertices in $S_x \cup S_y$) can be incident to exactly two edges. Therefore, for all but (at most) $2k$ trees of $G[V(G) \setminus (Z_v \cup \{v\})]$, every leaf of a tree in $G[V(G) \setminus (Z_v \cup \{v\})]$ must have at least one neighbor in $Z_v$. We now construct a bipartite graph $H$ with bipartition $(A = Z_v, B = D)$. We slightly abuse notation and assume that every component in $D$ corresponds to a vertex in $B$ and every vertex in $Z_v$ corresponds to a vertex in $A$. For every $D_i \in D$, $1 \leq i \leq p$, and for every $z \in Z_v$, $(D_i, z) \in E(H)$ if and only if there exists $u \in V(D_i)$ such that $(u, z) \in E(G)$. If $d_G(v) \geq 2k(k+2)+5k$ then the number of components $|D|$ is at least $2k(k+2)$ (taking into account the at most $k$ neighbors of $v$ which belong to components containing a cycle, the at most $2k$ neighbors of $v$ which belong to components with no neighbor in $Z_v$, and finally the at most $2k$ edges incident to $v$ and some vertex in $Z_v$). Consequently, $|D| \geq (k+2)|Z_v|$. We are now ready to state our main reduction rule.

Reduction Rule 7 If there exists a vertex $v \in V(G)$ such that $d_G(v) \geq 2k(k+2)+5k$ then let $D' \subseteq D$ and $Z_v' \subseteq Z_v$ be the sets obtained after applying Lemma 3 with $q = k+2$, $A = Z_v$, and $B = D$, such that $Z_v'$ has a $(k+2)$-expansion into $D'$ in $H$. Delete all the edges of the form $(u, v) \in E(G)$ such that $u \in D_i$ and $D_i \in D'$. Add two parallel edges between $v$ and every vertex in $Z_v'$.

Lemma 4 An instance $(G, S_x, S_y, k, \ell)$ of FVS-Bound is a yes-instance if and only if one of the $E+1$ reduced instances $(G', S_x', S_y', k-e, \ell-2(E-e))$, for $0 \leq e \leq E$, is a yes-instance.

Proof We show that no reduction rule removes possible reconfiguration sequences. This is trivially true for Reduction Rule 3. The vertices removed by Reduction Rules 1, 2, and 4 play different roles in converting a reconfiguration sequence for a reduced instance to a reconfiguration sequence for the original instance. As there is no cycle that can be destroyed only by a vertex removed from $O_{st}$ by Reduction Rule 1, 2, or 4, none of these vertices are needed. To account for the required removal (addition) of each such vertex in $S_x \setminus (S_y \setminus \ell)$, we remove all $r$ such vertices and decrease $\ell$ by $r$. We can choose to leave a vertex $v \in C_M$ in each solution in the sequence (with no impact on $\ell$) or to remove and then read $v$ to free up extra capacity, at a cost of incrementing $\ell$ by two; in the reduced instance we thus remove $v$ and either decrement $k$ or subtract
two from $\ell$. Since this choice can be made for each of these vertices, $E$ in total, we try to solve any of $E + 1$ instances $(G', S'_r, S'_r, k - e, \ell - 2(\ell - e))$ for $0 \leq e \leq \ell$.

For Reduction Rules 5 and 6 we show that the removed vertex $v$ is in $C_F$; since the cycles formed by $v$ must be handled by each solution in the sequence, the instance can be reduced by removing $v$ and decrementing $k$. By Fact 4 for $\Pi$ the set of acyclic graphs, there cannot be a cycle in $G[O_s \cup \{v\}]$ for any $v \in S_s \cup S_t \cup O_s$. Hence, whenever Reduction Rule 5 is applicable to some vertex $v$ then $v$ must be in $C_{st}$. The fact that $v \in C_F$ follows from the observation that every feedback vertex set must contain at least one vertex from the cycle corresponding to the loop at vertex $v$. In other words, a self-loop in a reduced instance simply corresponds to a path of degree-two vertices in the original instance. Hence, choosing $v$ into a solution to intersect this cycle is at least as good as choosing any other vertex. For Reduction Rule 6, $v \in C_F$ since any feedback vertex set not containing $v$ would have to contain at least $k + 1$ vertices, one for each cycle.

For Reduction Rule 7, it follows from Lemma 3 that for every vertex $z \in Z'_\nu \subseteq Z_v$ there are at least $k + 2$ internally vertex disjoint paths from $v$ to $z$ in $G$. Hence, any feedback vertex set of $G$ must either contain $v$ or all of $Z'_\nu$; as otherwise there are at least two paths from $v$ to some vertex in $Z'_\nu$ which form a cycle. Since we add two parallel edges between $v$ and every vertex in $Z'_\nu$ in $G'$, the same condition still holds in the reduced instance. $\Box$

Theorem 3 FVS- Reach and FVS- Bound parameterized by $k$ are in $\text{FPT}$. 

Proof After exhaustive application of all reduction rules, we know that every vertex in $V(G) \setminus (S_s \cup S_t)$ is incident to at least three edges (Reduction Rules 1 to 5) and every vertex in $V(G)$ is incident to at most $2k(k + 2) + 5k$ edges (Reduction Rules 6 and 7). Hence, applying Lemma 1, each reduced instance has $O(k^3)$ vertices and $O(k^3)$ edges. Reduction Rules 1 to 5 can clearly be applied in time polynomial in the size of $G$. The fact that we can apply Reduction Rules 6 and 7 in polynomial time follows from Lemmas 2 and 3. Finally, since the number of reduced instances is $E + 1 \leq |C_{st}| + 1 \leq k + 1$, as a consequence of Lemma 4, we have a reconfiguration kernel, as needed. $\Box$

4 Hardness Results

Before presenting a meta-hardness result in Sect. 4.1, we consider $\Pi$ to be the set of all edgeless graphs and prove the following lemma, which is a building block of the meta-theorem.

Lemma 5 VC- Bound parameterized by $\ell$, IS- Reach parameterized by $k$, and IS- Bound parameterized by $k + \ell$ are at least as hard as INDEPENDENT SET parameterized by $k$.

Proof Given an instance $(G, k)$ of INDEPENDENT SET, we construct a graph $G'$ by taking the disjoint union of $G$ and a $K_{k,k}$, a biclique with $k$ vertices on each side (Fig. 1). Formally, we let $V(G') = A \cup B \cup V_G$, where $A = \{a_i \mid 1 \leq i \leq k\}$, $B = \{b_i \mid 1 \leq i \leq k\}$, and $V_G = \{g_i \mid v_i \in V(G)\}$. The edge set of $G'$ is therefore

Springer
\[ G' = G + K_{4,4} \]

**Fig. 1** The graph \( G' \) consisting of the disjoint union of a graph \( G \) and a biclique \( K_{4,4} \)

\[ \text{defined as } E(G') = \{(g_i, g_j) | (v_i, v_j) \in E(G)\} \cup \{(a_i, b_j) | 1 \leq i, j \leq k\}. \]

We let \((G', S_s = V_G \cup A, S_t = V_G \cup B, |V_G| + k, \ell = 4k)\) be an instance of VC-Bound. In Fig. 1, \( S_s \) consists of the union of black and dark gray vertices and \( S_t \) consists of the union of black and light gray vertices.

If there is a reconfiguration sequence from \( S_s \) to \( S_t \), such a sequence cannot remove a vertex from \( A \) before adding all vertices in \( B \), as otherwise some edge would be left uncovered. In other words, whether we start from \( S_s \) and attempt to reach \( S_t \) or vice versa, one node, say \( S_f \), along any reconfiguration sequence between the two must consist of a vertex cover containing all vertices in \( A \cup B \). However, our maximum allowed capacity is \(|V_G| + k\) and \(|A \cup B| = 2k\). Therefore, \(|S_f \cap V_G| \leq |V_G| - k\).

Moreover, since \( S_f \) must be a vertex cover of \( G' \), we know that \( S_f \cap V_G \) must be a vertex cover of \( G'[V_G] \) (or equivalently \( G \)) of size at most \(|V_G| - k\). Consequently, \( I_f = V_G \setminus \{S_f \cap V_G\} \) is an independent set of size at least \( k \).

For the converse, we let \( I \) be an independent set in \( G \) (or equivalently in \( G'[V_G] \)) of size at least \( k \) and show how to obtain a reconfiguration sequence of length at most \( 4k \) from \( S_s \) to \( S_t \). The following sequence of \( 4k \) steps transforms \( S_s \) to \( S_t \): remove \( k \) vertices from \( I \), add all vertices in \( B \), remove all vertices in \( A \), and finally add back the \( k \) vertices in \( I \).

We have shown that \( G \) has an independent set of size at least \( k \) if and only if there is a path of length at most \( 4k \) from \( S_s \) to \( S_t \) in \( R_{\text{DEL}}^{\Pi}(G', |V(G)| + k) \). Since \(|V(G')| - (|V(G)| + k) = k\), this implies that \( G \) has an independent set of size at least \( k \) if and only if there is a path of length at most \( 4k \) from \( V(G') \setminus S_s \) to \( V(G') \setminus S_t \) in \( R_{\text{SUB}}^{\Pi}(G', k) \). Therefore, IS-Reach parameterized by \( k \) and IS-Bound parameterized by \( k + \ell \) are also at least as hard as INDEPENDENT SET parameterized by \( k \).

\[ \square \]

### 4.1 A Meta-Theorem

To generalize Lemma 5, we make use of the forbidden set characterization of hereditary properties. A \( \Pi \)-critical graph \( H \) is a minimal graph in the forbidden set \( F_{\Pi} \) that has at least two vertices; we use the fact that \( H \notin \Pi \), although the deletion of any vertex from \( H \) results in a graph in \( \Pi \). For convenience, we will refer to two of the vertices in
a $Π$-critical graph as terminals and the rest as internal vertices. We construct graphs from multiple copies of $H$. For a positive integer $c$, we let $H^*_c$ be the star graph obtained from each of $c$ copies $H_i$ of $H$ by identifying an arbitrary terminal $v_i$, $1 \leq i \leq c$, from each $H_i$; in $H^*_c$ vertices $v_1$ through $v_c$ are replaced with a vertex $w$, the gluing vertex of $v_1$ to $v_c$, to form a graph with vertex set $\cup_{1 \leq i \leq c}(V(H_i) \setminus \{v_i\}) \cup \{w\}$ and edge set $\cup_{1 \leq i \leq c}\{(u, v) \mid v \neq \{u, v_i\}\} \cup_{1 \leq i \leq c}\{(u, w) \mid (u, v_i) \in E(H_i)\}$. A terminal is non-identified if it is not used in forming a gluing vertex.

In Fig. 2, $H$ is a $K_3$ with terminals marked black and gray; $H^*_4$ is formed by identifying all the gray terminals to form $w$.

**Theorem 4** Let $Π$ be any hereditary property satisfying the following:

- For any two graphs $G_1$ and $G_2$ in $Π$, the graph obtained by their disjoint union is in $Π$.
- There exists a $Π$-critical graph $H \in \mathcal{F}_Π$ such that if $H^*_c$ is the graph obtained by identifying a terminal from each of $c$ copies of $H$, then the graph $R = H^*_c[V(H^*_c) \setminus \{u_1, u_2, \ldots, u_c\}]$ is in $Π$, where $u_1, u_2, \ldots, u_c$ are the non-identified terminals in the $c$ copies of $H$.

Then each of the following is at least as hard as $Π-$SUB($G, k$) parameterized by $k$:

1. $Π-$DEL- BOUND($G, S_s, S_t, k, \ell$) parameterized by $\ell$,
2. $Π-$SUB- REACH($G, S_s, S_t, k$) parameterized by $k$, and
3. $Π-$SUB- BOUND($G, S_s, S_t, k, \ell$) parameterized by $k + \ell$.

**Proof** Given an instance $(G, k)$ of $Π-$SUB and a $Π$-critical graph $H$ satisfying the hypothesis of the theorem, we form an instance $(G', S_s, S_t, |V(G)| + k, 4k)$ of $Π-$DEL- BOUND, with $G'$, $S_s$, and $S_t$ defined below. The graph $G'$ is the disjoint union of $G$ and a graph $W$ formed from $k^2$ copies of $H$, called $H_{i, j}$, where $1 \leq i, j \leq k$ and $\ell_{i, j}$ and $r_{i, j}$ are the terminals of $H_{i, j}$. We let $a_i$, $1 \leq i \leq k$, be the gluing vertex of $\ell_{i, 1}$ through $\ell_{i, k}$, and let $b_j$, $1 \leq j \leq k$, be the gluing vertex of $r_{1, j}$ through $r_{k, j}$, so that there is a copy of $H$ joining each $a_i$ and $b_j$. An example $W$ is shown in Fig. 3, where copies of $H$ are shown schematically as elongated gray ovals. We let $A = \{a_i \mid 1 \leq i \leq k\}$, $B = \{b_j \mid 1 \leq j \leq k\}$, $C = \{c \mid c \in V(W) \setminus (A \cup B)\}$, and $V_G = \{g_{i, j} \mid v_i \in V(G)\}$. The vertex and edge sets of $G'$ are therefore defined as $V(G') = A \cup B \cup C \cup V_G$ and $E(G') = \{(g_{i, j}, g_{k, l}) \mid (v_i, u_j) \in E(G)\} \cup E(W)$. We fix $S_s = A \cup V_G$ and $S_t = B \cup V_G$. Clearly, $|V(G')| = |V(G)| + 2k + k^2(|V(H)| - 2)$ and $|S_s| = |S_t| = |V(G)| + k$. Moreover, each of $V(G') \setminus S_t$ and $V(G') \setminus S_s$ induce a graph in $Π$, as each consists of $k$ disjoint copies of $H^*_k$ with one of the terminals removed from each $H$ in $H^*_k$.

Suppose the instance $(G', S_s, S_t, |V(G)| + k, 4k)$ of $Π-$DEL- BOUND is a yes-instance. As there is a copy of $H$ joining each vertex of $A$ to each vertex of $B$, before
removing $a \in A$ from $S_s$, the reconfiguration sequence must add all of $B$ to ensure that the complement of each intermediate set induces a graph in $\Pi$. Otherwise, the complement will contain at least one copy of $H$ as a subgraph and is therefore not in $\Pi$.

The capacity bound of $|V(G)| + k$ implies that the reconfiguration sequence must have removed from $S_s$ a subset $S' \subseteq V_G$ of size at least $k$ such that $V(G') \setminus (S_s \setminus S') = S' \cup B$ induces a subgraph in $\Pi$. Thus, $G[S'] \in \Pi$, and hence $\Pi$-SUB($G, k$) is a yes-instance.

Conversely, if the instance $(G, k)$ of $\Pi$-SUB is a yes-instance, then there exists $S' \subseteq V(G)$ such that $|S'| = k$ and $G[S'] \in \Pi$. We form a reconfiguration sequence between $S_s$ and $S_t$ by first deleting all vertices in $S'$ from $S_s$ to yield a set of size $|V(G)|$. $G'[V(G') \setminus (S_s \setminus S')]$ consists of the union of $G'[V'(G) \setminus S_s]$ and $G'[S'] = G[S']$, both of which are in $\Pi$.

Next we add one by one all vertices of $B$, delete one by one all vertices of $A$, and finally add back one by one each vertex in the set $S'$, resulting in a reconfiguration sequence of length $k + k + k + k = 4k$. It is clear that in every step, the complement of the set induces a graph in $\Pi$.

Thus we have showed that $\Pi$-SUB($G, k$) is a yes-instance if and only if there is a path of length at most $4k$ between $S_s$ and $S_t$ in $R^\Pi_{\text{DEL}}(G', |V(G)| + k)$. Since $|V(G')| - (|V(G)| + k) = k + k^2(|V(H)| - 2))$, this implies that $\Pi$-SUB($G, k$) is a yes-instance if and only if there is a path of length at most $4k$ between $V(G') \setminus S_s$ and $V(G') \setminus S_t$ in $R^\Pi_{\text{SUB}}(G', k + k^2(|V(H)| - 2))$. Therefore, $\Pi$-SUB-REACH parameterized by $k$ and $\Pi$-SUB-BOUND parameterized by $k + \ell$ are at least as hard as $\Pi$-SUB parameterized by $k$.

**Corollary 3** FVS-BOUND and OCT-BOUND are $\mathbf{W[1]}$-hard parameterized by $\ell$. IF-REACH and IBS-REACH are $\mathbf{W[1]}$-hard parameterized by $k$. IF-BOUND and IBS-BOUND are $\mathbf{W[1]}$-hard parameterized by $k + \ell$.

**Proof** It is known that for any hereditary property $\Pi$ that contains all edgeless graphs but not all cliques, $\Pi$-SUB parameterized by $k$ is $\mathbf{W[1]}$-hard [29].

For $\Pi$ the collection of all forests, $\Pi$ includes all edgeless graphs but does not include any clique of size three or more. Hence, the IF ($\Pi$-SUB) problem parameterized by solution size is $\mathbf{W[1]}$-hard. Moreover, as the disjoint union of any two forests is also a forest, $\Pi$ satisfies the first condition of Theorem 4. For the second condition,
given that the forbidden set $\mathcal{F}_T$ consists of all cycles, we let $H \in \mathcal{F}_T$ be a triangle; when we identify multiple triangles at a vertex, and remove another vertex of each of the triangles, we obtain a tree, which is in $\Pi$. Combining the fact that the hypothesis of Theorem 4 is satisfied with the $W[1]$-hardness of the IF problem, we know that FVS-Bound parameterized by $\ell$, IF-Reach parameterized by $k$, and IF-Bound parameterized by $k + \ell$ are all $W[1]$-hard.

The $W[1]$-hardness of OCT-Bound parameterized by $\ell$, IBS-Reach parameterized by $k$, and IBS-Bound parameterized by $k + \ell$ can be shown using similar arguments.

We obtain further results for interesting properties not covered by Theorem 4. Lemma 6 handles the collection of all cliques, which does not satisfy the first condition of the theorem, and the collection of all cluster graphs (disjoint unions of cliques), which satisfies the first condition but not the second. Moreover, as $\Pi$-SUB parameterized by $k$ is fixed-parameter tractable for $\Pi$ the collection of all cluster graphs [29] (also known as the CLUSTER SUBGRAPH problem [29]), Theorem 4 provides no lower bounds. It remains open whether Theorem 4 can be generalized even further to cover such properties.

**Lemma 6** CLIQUE-Reach and Cluster Subgraph-Reach parameterized by $k$ are $W[1]$-hard. CLIQUE-Bound and Cluster Subgraph-Bound parameterized by $k + \ell$ are $W[1]$-hard.

**Proof** We first give an FPT reduction from CLIQUE, known to be $W[1]$-hard, to CLUSTER SUBGRAPH-Bound. For $(G, k)$ an instance of CLIQUE, $V(G) = \{v_1, \ldots, v_n\}$, we form a graph consisting of four $K_k$‘s (with vertex sets $A$, $B$, $C$, and $D$) and a subgraph mimicking $G$ (with vertex set $X$), where there is an edge from each vertex in $X$ to each vertex in each $K_k$, and each of the subgraphs on the following vertex sets induce a $K_{2k}$: $A \cup B$, $A \cup C$, $B \cup D$, $C \cup D$. Formally, $V(G') = X \cup A \cup B \cup C \cup D$ and $E(G') = E_X \cup E_T \cup E_C$, where $X = \{x_1, \ldots, x_k\}$, $|A| = |B| = |C| = |D| = k$, $E_X = \{(x_i, x_j) \mid (v_i, v_j) \in E(G)\}$ corresponds to the edges in $G$, $E_T = \{(a, a') \mid a, a' \in A, a \neq a'\} \cup \{(b, b') \mid b, b' \in B, b \neq b'\} \cup \{(c, c') \mid c, c' \in C, c \neq c'\} \cup \{(d, d') \mid d, d' \in D, d \neq d'\}$ forms the $K_k$ cliques, and $E_C = \{(x, a), (x, b), (x, c), (x, d), (a, b), (a, c), (b, d), (c, d) \mid a \in A, b \in B, c \in C, d \in D, x \in X\}$ forms the connections among the vertex sets.

We let $(G', S_x, S_t, 2k, 6k)$ be an instance of CLUSTER SUBGRAPH-Bound, where $S_x = A \cup B$ and $S_t = C \cup D$. Clearly $|S_x| = |S_t| = 2k$ and both $S_x$ and $S_t$ induce cluster graphs (in fact cliques). We claim that $G$ has a clique of size $k$ if and only if there is a reconfiguration sequence of length $6k$ from $S_x$ to $S_t$.

If $G$ has a clique of size $k$, then there exists a subset $Y \subseteq X$ forming a clique of size $k$. We form a reconfiguration sequence of length $6k$ as follows: add the vertices in $Y$, remove the vertices in $A$, add the vertices in $D$, remove the vertices in $B$, add the vertices in $C$, and remove the vertices in $Y$, one by one. It is not hard to see that at every step in this sequence we maintain an induced clique in $G'$ of size greater than or equal to $2k$ (and hence a cluster subgraph). If there exists a reconfiguration sequence of length $6k$ from $S_x$ to $S_t$, we make use of the fact that no cluster subgraph contains an induced path of length three to show
that \( G \) has a clique of size \( k \). Observe that before adding any vertex of \( C \), we first need to remove (at least) all of \( B \) since otherwise we obtain an induced path of length three containing vertices in \( C, A, \) and \( B \), respectively. Similarly, we cannot add any vertex of \( D \) until we have removed all of \( A \). Therefore, before adding any vertex from \( S_t \), we first need to delete at least \( k \) vertices from \( S_s \). To do so without violating our minimum capacity of \( 2k \), at least \( k \) vertices must be added from \( X \). Since every vertex in \( X \) is connected to all vertices in \( S_s \) and \( S_t \), if any pair of those \( k \) vertices do not share an edge, we obtain an induced path on three vertices. Thus \( X \), and hence \( G \), must have a clique of size \( k \).

Since in our reduction \( S_s \) and \( S_t \) are cliques and every reconfiguration step maintains an induced clique in \( G' \) of size greater than or equal to \( 2k \), it follows that CLIQUE-REACH and CLUSTER SUBGRAPH-REACH parameterized by \( k \) and CLIQUE-BOUND and CLUSTER SUBGRAPH-BOUND parameterized by \( k + \ell \) are all \( \text{W[1]} \)-hard. \( \square \)

### 4.2 Unbounded Hitting Set

As neither DOMINATING SET nor its parametric dual can be expressed using hereditary graph properties, Theorem 4 is inapplicable; we instead use a construction specific to the problem in Lemma 7. This result in turn leads to Corollary 4, since DOMINATING SET can be cast as a hitting set of the family of closed neighborhoods of the vertices of a graph.

**Lemma 7** DS-REACH parameterized by \( k \) and DS-BOUND parameterized by \( k + \ell \) are \( \text{W[2]} \)-hard.

**Proof** We give a reduction from DOMINATING SET; for \((G, t)\) an instance of DOMINATING SET, we form \( G' \) as the disjoint union of two graphs \( G'_1 \) and \( G'_2 \).

We form \( G'_1 \) from \( t + 2 \) \((t + 1)\)-cliques \( C_0 \) (the outer clique) and \( C_1, \ldots, C_{t+1} \) (the inner cliques); \( V(C_0) = \{o_1, \ldots, o_{t+1}\} \) and \( V(C_i) = \{w(i,0), w(i,1), \ldots, w(i,t)\} \) for \( 1 \leq i \leq t + 1 \). The edge set of \( G'_1 \) contains not only the edges of the cliques but also \( \{(o_j, w(i,j)) | 1 \leq i \leq t + 1, 0 \leq j \leq t\} \); the graph to the left in Fig. 4 illustrates \( G'_1 \) for \( t = 2 \). Any dominating set that does not contain all vertices in the outer clique must contain a vertex from each inner clique.

To create \( G'_2 \), we first define \( G' \) to be the graph formed by adding a universal vertex to \( G \), where we assume without loss of generality that \( V(G) = \{v_1, \ldots, v_{|V(G)|}\} \).

![Fig. 4 Graphs used for the dominating set reduction](image-url)
We let $V(G'_2) = \bigcup_{0 \leq i \leq t} V(H_i)$, where $H_0, \ldots, H_t$ are $t + 1$ copies of $G^+$; we use $u_i$ to denote the universal vertex in $H_i$ and $v_{i,j}$ to denote the copy of $v_j$ in $H_i$, $1 \leq j \leq |V(G)|$, $0 \leq i \leq t$. The edge set consists of edges between each non-universal vertex $v_{(0,j)}$ in $H_0$ and, in each $H_i$, the universal vertex, its copy, and the copies of its neighbours in $G$, or more formally $E(G'_2) = \{(v_{0,j}, u_i) \mid 1 \leq j \leq |V(G)|, 1 \leq i \leq t\} \cup \{(v_{0,j}, v_{i,j}) \mid 1 \leq j \leq |V(G)|, 1 \leq i \leq t\} \cup \{(v_{0,j}, v_{i,k}) \mid 1 \leq j \leq |V(G)|, 1 \leq i \leq t, (v_j, v_k) \in E(G)\}$. The graph to the right in Fig. 4 illustrates part of $G'_2$, where universal vertices are shown in white and, for the sake of readability, the only edges outside of $G^+$ shown are those adjacent to a single vertex in $H_0$.

We form an instance $(G', D_x, D_t, 3t+2, 6t+4)$ of DS-BOUND, where $D_x = \{u_i \mid 0 \leq i \leq t\} \cup V(C_0)$ and $D_t = \{u_i \mid 0 \leq i \leq t\} \cup \{w_{i,i-1} \mid 1 \leq i \leq t + 1\}$. Both $D_x$ and $D_t$ are dominating sets, as each universal vertex $u_i$ dominates $H_i$ as well as $H_0$ and $V(G'_1)$ is dominated by the outer clique in $S$ and by one vertex from each inner clique in $T$. Clearly $|D_x| = |D_t| = 2t + 2$.

We claim that $G$ has a dominating set of size $t$ if and only if there is a reconfiguration sequence of length $6t + 4$ from $D_x$ to $D_t$. In $G'_1$, to remove any vertex from the outer clique, we must first add a vertex from each inner clique, for a total of $t + 1$ additions; since $k = 3t + 2$ and $|D_x| = 2t + 2$, this can only take place after $G'_2$ has been dominated using at most $t$ vertices. In $G'_2$, a universal vertex $u_i$ cannot be deleted until $H_i$ has been dominated. If $G$ can be dominated with $t$ vertices, then it is possible to add the dominating set in $H_0$ and remove all the universal vertices, thus making the required capacity available. If not, then no universal vertex $u_i$ can be removed without first adding at least $t + 1$ vertices to dominate $H_i$, for which there is not enough capacity. Therefore, there exists a reconfiguration sequence from $D_x$ to some $D'_x$ such that $D'_x \cap G'_2$ has $t$ vertices if and only if $G$ has a dominating set of size $t$. Moreover, the existence of a dominating set $D$ of size $t$ in $G$ implies a path of length $6t + 4$ from $D_x$ to $D_t$; we add $D$ in $H_0$, remove all universal vertices, apply all necessary reconfiguration steps to vertices in $G'_1$, add all universal vertices, and then remove $D$. Consequently, there exists a reconfiguration sequence from $D_x$ to $D_t$ in $6t + 4$ steps if and only if $G$ has a dominating set of size $t$.

The following is a result of there being a polynomial-time parameter-preserving reduction from DOMINATING SET to UNBOUNDED HITTING SET:

**Corollary 4** UHS-REACH parameterized by $k$ and UHS-BOUND parameterized by $k + \ell$ are $\mathsf{W[2]}$-hard.

5 Conclusions and Directions for Further Work

Our results constitute the first study of the parameterized complexity of reconfiguration problems. We give a general paradigm, the reconfiguration kernel, for proving fixed-parameter tractability, and provide hardness reductions that apply to problems associated with hereditary graph properties. It remains open whether Theorem 4 can be generalized to cover more hereditary graph properties by dropping one or both of the constraints. Also, as the reader might have noticed, none of our hardness results imply membership in $\mathsf{W[1]}$ nor $\mathsf{W[2]}$. It remains open where in the hierarchy of para-
meterized complexity classes the problems of reachability and bounded reachability belong.

Our result on cluster graphs (Lemma 6) demonstrates the existence of a problem that is fixed-parameter tractable [29], but whose reconfiguration version is \(W[1]\)-hard when parameterized by \(k\); this clearly implies that fixed-parameter tractability of the underlying problem does not guarantee fixed-parameter tractability of reconfiguration when parameterized by \(k\). Since there is unlikely to be a polynomial-sized kernel for the problem of determining whether a given graph has a cluster of size at least \(k\) [30], it is possible that an underlying problem having a polynomial-sized kernel is sufficient for the reconfiguration problem to be fixed-parameter tractable when parameterized by \(k\).

Our \textbf{FPT} algorithms for reconfiguration of \textsc{Bounded Hitting Set} and \textsc{Feedback Vertex Set} have running times of \(O^*(2^{O(k \log k)})\). Further work is needed to determine whether the running times can be improved to \(O^*(2^{O(k)})\), or whether these bounds are tight under the \textit{Exponential Time Hypothesis} [21,22].

We observe connections to another well-studied paradigm, local search [14], where the aim is to find an \textit{improved solution} at distance \(\ell\) of a given solution \(S\). Not surprisingly, as in local search, the problems we study turn out to be hard even in the parameterized setting when parameterized by \(\ell\). Other natural directions to pursue (as in the study of local search) are the parameterized complexity of reconfiguration problems in special classes of graphs and of reconfiguration problems on inputs other than graphs, as well as other parameterizations.

Since this work first appeared, there have been many subsequent developments [6,25–27,32,35,36]. For instance, Mouawad et al. [35] showed that \textsc{VC- Bound} parameterized by \(\ell\) remains \(W[1]\)-hard on bipartite graphs but is fixed-parameter tractable on graphs of bounded degree. Wrochna [42] showed that most problems considered in this work, in particular the reachability and bounded reachability variants of \textsc{Vertex Cover}, \textsc{Feedback Vertex Set}, and \textsc{Odd Cycle Transversal} (and their duals), are \textsc{PSPACE}-complete on graphs of bounded bandwidth, which also implies \textsc{PSPACE}-completeness on graphs of bounded treewidth. Subsequently, it was shown [36] that under parameterized complexity assumptions the bounded reachability variants of all aforementioned problems become fixed-parameter tractable parameterized by \(\ell\) and the treewidth of the input graph. Ito et al. [25,26] studied the parameterized complexity of a special version of the \textsc{IS- Reach} problem parameterized by \(k\) [25,26] and showed that it becomes fixed-parameter tractable on planar and bounded degree graphs. It was left open whether these results could be extended to graphs of bounded treewidth or other sparse graph classes. Lokshpanov et al. [32] answered this question positively for both \textsc{IS- Reach} and \textsc{DS- Reach} parameterized by \(k\). In a slightly different context, Bonsma et al. [6] and Johnson et al. [27] studied the parameterized complexity of graph recoloring. Both groups independently showed that, in contrast to graph vertex subset problems, the recoloring problem is fixed-parameter tractable when parameterized by \(\ell\).

\textbf{Acknowledgements}  The second author wishes to thank Marcin Kamiński for suggesting the examination of reconfiguration in the parameterized setting.
References

1. Bang-Jensen, J., Gutin, G.: Digraphs: theory, algorithms and applications. In: Springer Monographs in Mathematics. Springer-Verlag, London (2008)
2. Bodlaender, H.L.: A cubic kernel for feedback vertex set. In: Proceedings of the 24th Annual Conference on Theoretical Aspects of Computer Science, pp. 320–331 (2007)
3. Bonamy, M., Bousquet, N.: Recoloring bounded treewidth graphs. Electron. Notes Discrete Math. 44, 257–262 (2013)
4. Bonsma, P.: The complexity of rerouting shortest paths. In: Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science, pp. 222–233 (2012)
5. Bonsma, P., Cereceda, L.: Finding paths between graph colourings: PSPACE-completeness and super-polynomial distances. Theor. Comput. Sci. 410(50), 5215–5226 (2009)
6. Bonsma, P., Mouawad, A.E., Nishimura, N., Raman, V.: The complexity of bounded length graph recoloring and CSP reconfiguration. In: Proceedings of the 9th International Symposium on Parameterized and Exact Computation, pp. 110–121 (2014)
7. Cereceda, L.: Mixing graph colourings. Ph.D. thesis, London School of Economics (2007)
8. Cereceda, L., van den Heuvel, J., Johnson, M.: Mixing 3-colourings in bipartite graphs. Eur. J. Comb. 30(7), 1593–1606 (2009)
9. Cereceda, L., van den Heuvel, J., Johnson, M.: Finding paths between 3-colorings. J. Graph Theory 67(1), 69–82 (2011)
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshutanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer, Berlin (2015). doi: 10.1007/978-3-319-21275-3
11. Damaschke, P., Molokov, L.: The union of minimal hitting sets: parameterized combinatorial bounds and counting. J. Discrete Algorithms 7(4), 391–401 (2009)
12. Diestel, R.: Graph Theory. Springer, Berlin (2005). (Electronic Edition)
13. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer, New York (1997)
14. Fellows, M.R., Fomin, F.V., Lokshutanov, D., Rosamond, F., Saurabh, S., Villanger, Y.: Local search: is brute-force avoidable? J. Comput. Syst. Sci. 78(3), 707–719 (2012)
15. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
16. Gopalan, P., Kolaitis, P.G., Maneva, E.N., Papadimitriou, C.H.: The connectivity of Boolean satisfiability: computational and structural dichotomies. SIAM J. Comput. 38(6), 2330–2355 (2009)
17. Guo, J., Gramm, J., Hüffner, F., Niedermeier, R., Wernicke, S.: Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization. J. Comput. Syst. Sci. 72(8), 1386–1396 (2006)
18. Haas, R., Seyffarth, K.: The k-dominating graph. Graphs Comb. 30(3), 609–617 (2014)
19. Haddadan, A., Ito, T., Mouawad, A.E., Nishimura, N., Ono, H., Suzuki, A., Tebbal, Y.: The complexity of dominating set reconfiguration. In: Proceedings of the 14th Algorithms and Data Structures Symposium (to appear) (2015)
20. Hear, R.A., Demaine, E.D.: PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. Theor. Comput. Sci. 343(1–2), 72–96 (2005)
21. Impagliazzo, R., Paturi, R.: The complexity of k-SAT. In: Proceedings of the 14th Annual IEEE Conference on Computational Complexity, pp. 237–240 (1999)
22. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. Syst. Sci. 63(4), 512–530 (2001)
23. Ito, T., Demaine, E.D., Harvey, N.J.A., Papadimitriou, C.H., Sideri, M., Uehara, R., Uno, Y.: On the complexity of reconfiguration problems. Theor. Comput. Sci. 412(12–14), 1054–1065 (2011)
24. Ito, T., Kamiński, M., Demaine, E.D.: Reconfiguration of list edge-colorings in a graph. Discrete Appl. Math. 160(15), 2199–2207 (2012)
25. Ito, T., Kamiński, M., Ono, H.: Fixed-parameter tractability of token jumping on planar graphs. In: Algorithms and Computation, Lecture Notes in Computer Science, Springer International Publishing, pp. 208–219 (2014)
26. Ito, T., Kamiński, M., Ono, H., Suzuki, A., Uehara, R., Yamanaka, K.: On the parameterized complexity for token jumping on graphs. In: Proceedings of the 11th Annual Conference on Theory and Applications of Models of Computation. Lecture Notes in Computer Science, vol. 8402, pp. 341–351. Springer International Publishing (2014)
27. Johnson, M., Kratsch, D., Kratsch, S., Patel, V., Paulusma, D.: Finding shortest paths between graph colourings. In: Proceedings of the 9th International Symposium on Parameterized and Exact Computation, pp. 221–233 (2014)
28. Kamiński, M., Medvedev, P., Milanič, M.: Shortest paths between shortest paths. Theor. Comput. Sci. 412(39), 5205–5210 (2011)
29. Khot, S., Raman, V.: Parameterized complexity of finding subgraphs with hereditary properties. Theor. Comput. Sci. 289(2), 997–1008 (2002)
30. Kratsch, S., Pilipczuk, M., Rai, A., Raman, V.: Kernel lower bounds using co-nondeterminism: finding induced hereditary subgraphs. Trans. Comput. Theory 7(1), 4:1–4:18 (2014)
31. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. 20(2), 219–230 (1980)
32. Lokshtanov, D., Mouawad, A.E., Panolan, F., Ramanujan, M., Saurabh, S.: Reconfiguration on sparse graphs. In: Proceedings of the Algorithms and Data Structures Symposium (to appear) (2015)
33. Misra, N., Narayanaswamy, N., Raman, V., Shankar, B.S.: Solving min ones 2-SAT as fast as vertex cover. Theor. Comput. Sci. 506, 115–121 (2013)
34. Mouawad, A.E., Nishimura, N., Pathak, V., Raman, V.: Shortest reconfiguration paths in the solution space of Boolean formulas. In: Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming, vol. 9134, pp. 985–996 (2015)
35. Mouawad, A.E., Nishimura, N., Raman, V.: Vertex cover reconfiguration and beyond. In: Proceedings of the 25th International Symposium on Algorithms and Computation, pp. 452–463 (2014)
36. Mouawad, A.E., Nishimura, N., Raman, V., Wrochna, M.: Reconfiguration over tree decompositions. In: Proceedings of the 9th International Symposium on Parameterized and Exact Computation, pp. 246–257 (2014)
37. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford (2006)
38. Raman, V., Saurabh, S.: Parameterized algorithms for feedback set problems and their duals in tournaments. Theor. Comput. Sci. 351(3), 446–458 (2006)
39. Suzuki, A., Mouawad, A.E., Nishimura, N.: Reconfiguration of dominating sets. In: Proceedings of the 20th International Computing and Combinatorics Conference, pp. 405–416 (2014)
40. Thomassé, S.: A 4k^2 kernel for feedback vertex set. ACM Trans. Algorithms 6(2), 32:1–32:8 (2010)
41. van den Heuvel, J.: The complexity of change. Surv. Comb. 409, 127–160 (2013)
42. Wrochna, M.: Reconfiguration in bounded bandwidth and treedepth. CoRR (2014). ArXiv:1405.0847