Dynamic Parameterized Problems *

R. Krithika, Abhishek Sahu, and Prafullkumar Tale

The Institute of Mathematical Sciences, HBNI, Chennai, India.
{rkrithika|asahu|pptale}@imsc.res.in

Abstract

We study the parameterized complexity of various graph theoretic problems in the dynamic framework where the input graph is being updated by a sequence of edge additions and deletions. Vertex subset problems on graphs typically deal with finding a subset of vertices having certain properties. In real world applications, the graph under consideration often changes over time and due to this dynamics, the solution at hand might lose the desired properties. The goal in the area of dynamic graph algorithms is to efficiently maintain a solution under these changes. Recomputing a new solution on the new graph is an expensive task especially when the number of modifications made to the graph is significantly smaller than the size of the graph. In the context of parameterized algorithms, two natural parameters are the size k of the symmetric difference of the edge sets of the two graphs (on n vertices) and the size r of the symmetric difference of the two solutions. We study the Dynamic Π-Deletion problem which is the dynamic variant of the classical Π-Deletion problem and show NP-hardness, fixed-parameter tractability and kernelization results. For specific cases of Dynamic Π-Deletion such as Dynamic Vertex Cover and Dynamic Feedback Vertex Set, we describe improved algorithms and linear kernels. Specifically, we show that Dynamic Vertex Cover has a deterministic algorithm with $1.0822^k n^{\mathcal{O}(1)}$ running time and Dynamic Feedback Vertex Set has a randomized algorithm with $1.6667^k n^{\mathcal{O}(1)}$ running time. We also show that Dynamic Connected Feedback Vertex Set can be solved in $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ time. For each of Dynamic Connected Vertex Cover, Dynamic Dominating Set and Dynamic Connected Dominating Set, we describe an algorithm with $2^k n^{\mathcal{O}(1)}$ running time and show that this is the optimal running time (up to polynomial factors) assuming the Set Cover Conjecture.

1 Introduction

Graphs are discrete mathematical structures that represent pairwise relations between objects. Due to their tremendous power to model real world systems, many problems of practical interest can be represented as problems on graphs. Consequently, the design of algorithms on graphs is of major importance in computer science. Applications that employ graph algorithms typically involve large graphs that change over time. A natural goal in this setting is to design algorithms that efficiently maintain a solution under these changes. That is, given a graph G and a solution S, one searches for a solution S' that is as close as possible to S in a graph G' that can be obtained from G by making at most k changes. We only consider instances where the graphs under consideration have the same vertex set. Formally, a dynamic version of a graph theoretic problem is a quintuple (G, G', S, k, r) where G and G' are graphs on the same vertex set. The size of the symmetric difference of the edge sets of G and G' is upper bounded by k and S is a solution (not necessarily optimal) for G. The task is to determine whether there is a solution S' (also not necessarily optimal) for G' such that the symmetric difference of S and S' is at most r.

Dynamic problems have been recently studied in the parameterized complexity framework [AKEF$^+$15, DEF$^+$14, HN13, GGJ$^+$17, AKCE$^+$17]. In parameterized complexity, each problem

*A preliminary version of this work appears in the proceedings of the 11th International Symposium on Parameterized and Exact Computation (IPEC 2016).

instance is associated with a non-negative integer called *parameter*. A common parameter is a bound on the size of an optimum solution to the problem instance. A problem is said to be *fixed-parameter tractable* (FPT) with respect to parameter ℓ if it can be solved in $f(\ell)n^{\mathcal{O}(1)}$ time for some computable function f, where n is the input size. For convenience, the running time $f(\ell)n^{\mathcal{O}(1)}$ where f grows super polynomially with ℓ is denoted as $\mathcal{O}^*(f(\ell))$. In order to classify parameterized problems as being FPT or not, the W-hierarchy: $\mathsf{FPT} \subseteq \mathsf{W}[1] \subseteq \mathsf{W}[2] \subseteq \cdots \subseteq \mathsf{XP}$ is defined. It is believed that the subset relations in this sequence are all strict and a parameterized problem that is hard for some complexity class above FPT in this hierarchy is unlikely to be FPT. Such a hard problem is said to be *fixed-parameter intractable*. A *kernelization algorithm* is a polynomial-time algorithm that transforms an arbitrary instance (I, l) of the problem to an equivalent instance (I', l') of the same problem such that $|I'| + l' = g(l)$ for some computable function g. The instance (I', l') is called a *kernel* and if $g(l) = l^{\mathcal{O}(1)}$, then it is called a *polynomial kernel* and we say that the problem admits a polynomial kernel. Further details on parameterized algorithms can be found in [CFL+15, DF13, FG06].

Two relevant parameters for dynamic problem instances are the *edit parameter* k and the *distance parameter* r. Parameterized complexity results for the dynamic versions of various problems with respect to these parameters are known [DEF+14, AKEF+15]. In this work, we revisit several classical parameterized problems in the dynamic setting. Let Π be a fixed collection of graphs. Given a graph G and an integer l, the Π-Deletion problem is to determine if G has a set $S \subseteq V(G)$ of vertices with $|S| \leq l$ such that $G - S \in \Pi$. Π-Deletion is an abstraction of various problems on graphs. Due to a generic result by Lewis and Yannakakis [LY80], Π-Deletion is known to be NP-hard for most choices of Π. Hence, it has been extensively studied in various algorithmic realms. We refer to the dynamic version of this problem as Dynamic Π-Deletion and show NP-hardness, fixed-parameter tractability and kernelization results. For the specific cases of Dynamic Π-Deletion such as Dynamic Vertex Cover and Dynamic Feedback Vertex Set, we describe improved FPT algorithms and linear kernels with respect to the edit parameter. We also show that Dynamic Connected Feedback Vertex Set is FPT with respect to the edit parameter. Further, we describe FPT algorithms for Dynamic Connected Vertex Cover, Dynamic Dominating Set and Dynamic Connected Dominating Set with improved running times for the same parameterization. We show that for each of these problems, this running time is the best possible (up to polynomial factors) under the *Set Cover Conjecture* (defined subsequently). Table 1 summarizes our results along with the running time bounds known for these problems.

Dynamic Problem	Parameter k	Parameter r
Vertex Cover	1.0822^k $\mathcal{O}(k)$ kernel	1.2738^r $\mathcal{O}(r^2)$ kernel
Connected Vertex Cover	4^k [AKEF+15], 2^k ‡ No $k^{\mathcal{O}(1)}$ size kernel	W[2]-hard [AKEF+15]
Feedback Vertex Set	1.6667^k (randomized) $\mathcal{O}(k)$ kernel	3.592^r, 3^r (randomized) $\mathcal{O}(r^2)$ kernel
Connected Feedback Vertex Set	$2^{\mathcal{O}(k)}$ No $k^{\mathcal{O}(1)}$ size kernel	W[2]-hard
Dominating Set	$2^{\mathcal{O}(k^2)}$ [DEF+14], 2^k ‡ No $k^{\mathcal{O}(1)}$ size kernel	W[2]-hard [DEF+14]
Connected Dominating Set	4^k [AKEF+15], 2^k ‡ No $k^{\mathcal{O}(1)}$ size kernel	W[2]-hard [AKEF+15]

Table 1: Summary of known and new results for different dynamic parameterized problems. All running time bounds are specified by ignoring polynomial factors. ‡ denotes that the running time is optimal under the Set Cover Conjecture.

Some of our algorithms involve a reduction to the Group Steiner Tree problem or to the Set Cover problem. We now define these problems. Given a graph G, an integer p and a family $\mathcal{F}$ of subsets of $V(G)$, the Group Steiner Tree problem is the task of determining whether G contains a tree on at most p vertices that contains at least one vertex from each set in $\mathcal{F}$. This problem is known to admit an algorithm with $\mathcal{O}^*(2^{|\mathcal{F}|})$ running time [MPR+12]. Given a family $\mathcal{F}$ of subsets of a universe U and a positive integer ℓ, the Set Cover problem is the task of determining whether there is a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ of size at most ℓ such that $\bigcup_{X \in \mathcal{F}'} X = U$. Set Cover can be solved in $\mathcal{O}^*(2^{|U|})$ time by a dynamic programming routine [FKW05] and the *Set Cover Conjecture* states that no $\mathcal{O}^*((2-\epsilon)^{|U|})$ time algorithm exists for any $\epsilon > 0$ [CDL+16]. Our lower bound results are based on this conjecture. To show certain kernelization hardness results, we use the fact that Set Cover parameterized by the universe size does not have polynomial kernels unless $\mathsf{NP} \subseteq \mathsf{coNP}/\text{poly}$ [DLS09].

Preliminaries: Parameterized complexity terminology and definitions not stated here can be found in [CFL+15]. Graph theoretic notation/definitions not given explicitly here can be found in [Die05]. All graphs considered in this paper are finite, undirected, unweighted and simple. For a graph G, $V(G)$ and $E(G)$ denote the set of vertices and edges respectively. The symmetric difference of two subsets $S, S' \subseteq V(G)$, denoted by $d_v(S, S')$, is defined as the size of the set $(S \setminus S') \cup (S' \setminus S)$. For two graphs G and G' on the same vertex set, $d_e(G, G')$ denotes the size of the symmetric difference of $E(G)$ and $E(G')$. For a vertex u in a graph G, its neighbourhood $N_G(u)$, is set of all the vertices adjacent to it and its closed neighbourhood $N_G[u]$ is the set $N_G(u) \cup \{u\}$. This definition is extended to subsets of vertices as $N_G[S] = \bigcup_{v \in S} N_G[v]$ and $N_G(S) = N_G[S] \setminus S$ where $S \subseteq V(G)$. The subscript in the notation for neighbourhood is omitted if the graph under consideration is clear from the context. For a set of edges E', $V(E')$ denotes the union of the endpoints of the edges in E'. For a set $S \subseteq V(G)$, $G[S]$ and $G - S$ denote the subgraphs of G induced by the set S and $V(G) \setminus S$ respectively. The *contraction* operation of an edge $e = uv$ in G adds a new vertex w adjacent to the vertices that are adjacent to either u or v and then deletes u and v. That is, the contraction of e in G results in a graph G' with $V(G') = (V(G) \cup \{w\}) \setminus \{u, v\}$ and $E(G') = \{xy \mid x, y \in V(G) \setminus \{u, v\} \text{ and } xy \in E(G)\} \cup \{wx \mid x \in N_G(u) \cup N_G(v)\}$. An *independent set* is a set of pairwise non-adjacent vertices and a *clique* is a set of pairwise adjacent vertices. We also call an edgeless graph as an independent set and a complete graph as a clique. A *forest* is a graph with no cycles.

2 Complexity of Dynamic Π-Deletion

A graph property Π is a collection of graphs. Π is said to be *hereditary* if for any graph in Π, all of its induced subgraphs are in Π too. The membership testing problem for Π is the task of determining if a graph is in Π or not. For most natural choices of Π, the Π-Deletion problem is NP-hard [LY80] and interesting dichotomy results are known in the parameterized complexity framework [Cai96, KR02]. We formally define its dynamic variant referred to as Dynamic Π-Deletion as follows. Let I_n denote the graph on n vertices with no edges and K_n denote the complete graph on n vertices.

> Dynamic Π-Deletion
> **Input:** Graphs G, G' on the same vertex set, a set $S \subseteq V(G)$ such that $G - S \in \Pi$ and integers k, r with $d_e(G, G') \leq k$.
> **Question:** Does there exist $S' \subseteq V(G')$ with $d_v(S, S') \leq r$ such that $G' - S' \in \Pi$?

Observe that if Π-Deletion is in NP then so is Dynamic Π-Deletion. We are now ready to state our first result.

Theorem 1. *Let Π be a graph property that includes all independent sets or all cliques. Then,* Π-Deletion *reduces to* Dynamic Π-Deletion *in polynomial time.*

Proof. Let (H, l) be an instance of Π-Deletion where H is a graph on n vertices. We reduce (H, l) to the instance $(G, G' = H, S = \emptyset, k, r = l)$ of Dynamic Π-Deletion as follows. If Π includes all independent sets, then $G = I_n$ and $k = |E(H)|$. Otherwise, $G = K_n$ and $k = \binom{|V(H)|}{2} - |E(H)|$. In both cases, by the property of Π, $G - S \in \Pi$. Also, the vertex sets of H, G and G' are the same. Then, for a set $S' \subseteq V(H)$, we have $H - S' \in \Pi$ if and only if $G' - S' \in \Pi$ such that $d_v(S, S') = |S'|$. □

As a consequence of Theorem 1, we have the following hardness result.

Corollary 2. *The following results hold for a property Π that includes all independent sets or all cliques.*

- *If* Π-Deletion *is* NP*-hard, then* Dynamic Π-Deletion *is* NP*-hard.*
- *If* Π-Deletion *parameterized by solution size is fixed-parameter intractable then* Dynamic Π-Deletion *parameterized by* r *is fixed-parameter intractable.*
- *If* Π-Deletion *is* NP*-complete and does not admit a polynomial kernel when parameterized by solution size then* Dynamic Π-Deletion *parameterized by* r *does not admit a polynomial kernel.*

Proof. The NP-hardness and the fixed-parameter intractability results follow straightaway from Theorem 1. If Π-Deletion is NP-complete, then Dynamic Π-Deletion reduces to Π-Deletion in polynomial time. Therefore, if Dynamic Π-Deletion parameterized by r admits a polynomial kernel, such a kernel can be transformed to a polynomial kernel for Π-Deletion parameterized by solution size using this reduction and the reduction described in Theorem 1.Thus, the claimed kernelization hardness follows too. □

The following observation shows that for many choices of Π, to solve Dynamic Π-Deletion, it suffices to look for a new solution that contains the current solution.

Observation 3. *Let Π be a hereditary property. If S' is a solution to the instance (G, G', S, k, r), then there is another solution S'' with $d_v(S, S'') \leq d_v(S, S')$ and $S \subseteq S''$.*

Proof. As $G' - S' \in \Pi$ and Π is hereditary, it follows that $G' - (S \cup S') \in \Pi$ as well. Further, $d_v(S, S \cup S') = |S' \setminus S| \leq d_v(S, S')$. □

Now, we proceed to show certain tractable cases of Dynamic Π-Deletion.

Theorem 4. *Let Π be a hereditary property whose membership testing problem is polynomial-time solvable. Then,* Dynamic Π-Deletion *reduces to* Π-Deletion *in polynomial time.*

Proof. Consider an instance (G, G', S, k, r) of Dynamic Π-Deletion. The task is to determine if G' has a solution S' with $d_v(S, S') \leq r$. If $G' - S \in \Pi$, then S is the required solution S'. Otherwise, from Observation 3, we may assume that S' contains S. Let H denote the graph $G' - S$. Then, $H - (S' \setminus S) \in \Pi$. Therefore, for a set $T \subseteq V(H)$, we have $H - T \in \Pi$ if and only if $G' - (S \cup T) \in \Pi$ such that $d_v(S, S \cup T) = |T|$. □

Now, the following claim holds.

Corollary 5. *Let Π be a hereditary property whose membership testing problem is polynomial-time solvable. If* Π-Deletion *is* FPT *with respect to the solution size l as parameter, then* Dynamic Π-Deletion *is* FPT *both with respect to r as parameter and with respect to k as parameter.*

Proof. Consider an instance (G, G', S, k, r) of Dynamic Π-Deletion. Suppose Π-Deletion has an algorithm with $\mathcal{O}^*(f(l))$ running time. Then, from Theorem 4, there is an algorithm $\mathcal{A}$ solving (G, G', S, k, r) in $\mathcal{O}^*(f(r))$ time. Thus, the problem is FPT when parameterized by r. Let $\tilde{E}$ denote the set $(E(G') \setminus E(G)) \cup (E(G) \setminus E(G'))$. Let T be a set of vertices of G' that contains at least one endpoint of each edge in $\tilde{E}$. Clearly, T has at most k vertices. As Π is hereditary and $G - S \in \Pi$, it follows that $G' - (S \cup T) \in \Pi$. Now, if $r \geq k$, then $S \cup T$ is the required solution S'. Otherwise, the algorithm $\mathcal{A}$ solving Dynamic Π-Deletion runs in $\mathcal{O}^*(f(k))$ time. Hence, the problem is FPT when parameterized by k. □

Finally, we move on to kernelization results for the problem.

Corollary 6. *Let Π be a hereditary property whose membership testing problem is polynomial-time solvable. The following results hold when* Π-Deletion *parameterized by the solution size l admits a kernel with $p(l)$ vertices and $q(l)$ edges.*

- *If Π includes all independent sets, then* Dynamic Π-Deletion *admits a kernel with $2p(r) \leq 2p(k)$ vertices and $q(r) \leq q(k)$ edges.*
- *If Π includes all cliques, then* Dynamic Π-Deletion *admits a kernel with $2p(r) \leq 2p(k)$ vertices and $q(r) + p(r)^2 \leq q(k) + p(k)^2$ edges.*

Proof. Consider an instance (G, G', S, k, r) of Dynamic Π-Deletion. If $G' - S \in \Pi$ or $r \geq k$, the output of the kernelization algorithm is $(K_1, \emptyset, K_1, 0, 0)$ (with constant size) which is a trivial yes-instance of Dynamic Π-Deletion. Suppose $G' - S \notin \Pi$ and $r < k$. Let (H', r') be the kernelized instance of (H, r), the instance of Π-Deletion obtained from Theorem 4. Then, $(H'', H', \emptyset, |E(H')|, r')$ is the kernelized instance of (G, G', S, k, r) where $H'' = I_{|V(H')|}$ if Π includes all independent sets and $H'' = K_{|V(H')|}$ if Π includes all cliques. Hence, the claimed bounds on the kernel size follow. □

Remark: A property Π is called *interesting* if the number of graphs in Π and the number of graphs not in Π are unbounded. Any hereditary property that is interesting either contains all independent sets or contains all cliques. Thus, all the above results hold for such properties. In particular, the results of this section hold for the dynamic variants of classical problems like Vertex Cover, Feedback Vertex Set, Odd Cycle Transversal and Split Vertex Deletion.

3 Dynamic Vertex Cover

A *vertex cover* is a set of vertices that has at least one endpoint from every edge and Dynamic Vertex Cover is formally defined as follows.

Dynamic Vertex Cover **Input:** Graphs G, G' on the same vertex set, a vertex cover S of G and integers k, r such that $d_e(G, G') \leq k$. **Question:** Does there exist a vertex cover S' of G' such that $d_v(S, S') \leq r$?

Clearly, Dynamic Vertex Cover is Dynamic Π-Deletion where Π is the set of all independent sets. As Vertex Cover, the problem of determining if a graph has a vertex cover of size l, is NP-hard, its dynamic version is NP-hard too by Theorem 1. In [AKEF+15], the authors claim that Dynamic Vertex Cover is W[1]-hard with respect to $k + r$ as parameter by a reduction from Independent Set parameterized by the solution size. However, the reduction is incorrect and the fixed-parameter intractability does not follow. Vertex Cover parameterized by the solution size l admits a kernel with at most $2l$ vertices [CKX10] and an algorithm with $\mathcal{O}^*(1.2738^l)$ running time [CKJ01]. By Theorem 4 and Corollaries 5 and 6, these results extend to Dynamic Vertex Cover as well. In particular, the following results hold.

- Dynamic Vertex Cover can be solved in $\mathcal{O}^*(1.2738^r)$ time and in $\mathcal{O}^*(1.2738^k)$ time.
- Dynamic Vertex Cover admits a kernel with at most $4r$ vertices and $O(r^2)$ edges.
- Dynamic Vertex Cover admits a kernel with at most $4k$ vertices and $O(k^2)$ edges.

We now improve over these results by describing a linear kernel and a faster algorithm for the problem when parameterized by k. First, we describe the linear kernelization.

Theorem 7. Dynamic Vertex Cover *admits a kernel with at most* $2k$ *vertices and* k *edges.*

Proof. Consider an instance (G, G', S, k, r) of Dynamic Vertex Cover. By Observation 3, it suffices to search for a solution S' that contains S. As $d_e(G, G') \leq k$, we have $|E(G') \setminus E(G)| \leq k$. Also, edges in $E(G) \setminus E(G')$ do not affect the solution. Let H be the graph with $V(H) = V(E(G') \setminus E(G))$ and $E(H) = E(G') \setminus E(G)$. Then, H has at most $2k$ vertices and k edges. Further, (G, G', S, k, r) is a yes-instance of Dynamic Vertex Cover if and only if (H, r) is a yes-instance of Vertex Cover. Then, from Corollary 6, it suffices to output a linear kernel of the instance (H, r). We apply the following standard preprocessing on H.

Reduction Rule 3.1. *Delete isolated vertices.*

Reduction Rule 3.2. *If there is a vertex* v *of degree* 1*, add its neighbour* u *into the solution and decrease* r *by 1. Delete* u *and* v *from the graph.*

Let H' denote the resultant graph on which these rules are no longer applicable and r' denote the budget. As the rules are safe (i.e., they preserve minimum vertex covers), we have the following equivalence: (H, r) is a yes-instance of Vertex Cover if and only if (H', r') is a yes-instance of Vertex Cover. Then, as the minimum degree of H' is at least 2, we have $|E(H')| \geq |V(H')|$. As $|E(H')| \leq k$, it follows that $|V(H')| \leq k$. Thus, from Corollary 6, the kernelized instance corresponding to (G, G', S, k, r) is $(I_{|V(H')|}, H', \emptyset, k = |E(H')|, r')$. □

Next, we describe an algorithm (faster than the $\mathcal{O}^*(1.2738^k)$ time algorithm) for the problem when parameterized by k.

Theorem 8. Dynamic Vertex Cover *can be solved in* $\mathcal{O}^*(1.0822^k)$ *time.*

Proof. Consider an instance (G, G', S, k, r) of Dynamic Vertex Cover. By Observation 3, it suffices to search for a solution S' that contains S. Let H be the graph with $V(H) = V(E(G') \setminus E(G))$ and $E(H) = E(G') \setminus E(G)$. Then, H has at most $2k$ vertices and k edges and it suffices to solve the instance (H, r) of Vertex Cover. We first apply Reduction Rules 3.1 and 3.2 on H as long as they are applicable. Then, $|V(H)| \leq |E(H)| \leq k$. It is known that a minimum vertex cover of a graph with m edges can be obtained in $\mathcal{O}^*(1.0822^m)$ time [Bei99]. Thus, an $\mathcal{O}^*(1.0822^k)$ algorithm follows. □

4 Dynamic Connected Vertex Cover

A *connected vertex cover* is a vertex cover that induces a connected subgraph and Dynamic Connected Vertex Cover is defined as follows.

> Dynamic Connected Vertex Cover
> **Input:** Graphs G, G' on the same vertex set, a connected vertex cover S of G and integers k, r such that $d_e(G, G') \leq k$.
> **Question:** Does there exist a connected vertex cover S' of G' such that $d_v(S, S') \leq r$?

The problem is NP-complete, W[2]-hard when parameterized by r and admits an $\mathcal{O}^*(4^k)$ algorithm by a reduction to finding a minimum weight Steiner tree [AKEF+15]. We describe an $\mathcal{O}^*(2^k)$

algorithm by a reduction to finding a group Steiner tree. First, we observe a property of a solution to an instance of Dynamic Connected Vertex Cover analogous to Observation 3. Consider an instance (G, G', S, k, r) of Dynamic Connected Vertex Cover. Observe that G' must be connected, otherwise, (G, G', S, k, r) is a no-instance. Then, any set that contains a connected vertex cover is also a connected vertex cover. As $d_v(S, S' \cup S) = |S' \setminus S| \leq d_v(S, S')$, the following claim holds.

Observation 9. *If S' is a connected vertex cover of G', then $S' \cup S$ is also a connected vertex cover of G' with $d_v(S, S' \cup S) \leq d_v(S, S')$.*

Now, we prove the main result of this section.

Theorem 10. Dynamic Connected Vertex Cover *can be solved in $\mathcal{O}^*(2^k)$ time.*

Proof. Consider an instance (G, G', S, k, r) of Dynamic Connected Vertex Cover. By Observation 9, we can assume that the required solution S' contains S. Observe that $G'[S]$ is not necessarily connected and the edges in G' that are not covered by S are those edges in $E' = (E(G') \setminus E(G)) \cap E(G' - S)$. Now, we show a reduction to finding a group Steiner tree. Contract each component of $G'[S]$ to a single vertex. Let H denote the resulting graph and let $X = V(H) \setminus V(G')$. Construct an instance $(H, |X| + r, \mathcal{F})$ of Group Steiner Tree where $\mathcal{F} = \{\{u, v\} \mid uv \in E'\} \cup \{\{x\} \mid x \in X\}$. We claim that (G, G', S, k, r) is a yes-instance of Dynamic Connected Vertex Cover if and only if $(H, |X| + r, \mathcal{F})$ is a yes-instance of Group Steiner Tree.

Suppose there is a connected vertex cover S' of G' such that $d_v(S, S') \leq r$ and $S \subseteq S'$. As $G'[S']$ is connected, it follows that $H[X \cup (S' \cap V(G' - S))]$ is also connected. Moreover, as $|S' \cap V(G' - S)| \leq r$, it follows that the spanning tree of $H[X \cup (S' \cap V(G' - S))]$ is of size at most $|X| + r$. Hence $(H, |X| + r, \mathcal{F})$ is a yes-instance of Group Steiner Tree. Conversely, suppose $(H, |X| + r, \mathcal{F})$ is a yes-instance of Group Steiner Tree. Let T denote the solution tree of H. Then, $X \subseteq V(T)$ and $|V(T) \setminus X| = |V(T) \cap V(G' - S)| \leq r$. Define $S' = S \cup (V(T) \cap V(G' - S))$. The size of S' is at most $|S| + r$. Further, $G'[S']$ is connected as S' is obtained from the vertices of T. Also, for every edge in E', T contains at least one of its endpoints. Thus, S' is the desired connected vertex cover of G'. As the sum of the number of components of $G'[S]$ and the size of E' is upper bounded by $k+1$, it follows that $|\mathcal{F}| \leq k+1$. Thus, the Group Steiner Tree algorithm of [MPR+12] runs in $\mathcal{O}^*(2^k)$ time. □

Next, we show a lower bound on the running time of an algorithm that solves Dynamic Connected Vertex Cover assuming the Set Cover Conjecture. We do so by a reduction from Set Cover to Dynamic Connected Vertex Cover.

Theorem 11. Dynamic Connected Vertex Cover *does not admit an algorithm with $\mathcal{O}^*((2-\epsilon)^k)$ running time for any $\epsilon > 0$ assuming the Set Cover Conjecture.*

Proof. Consider an instance $(U, \mathcal{F}, \ell)$ of Set Cover where $U = \{u_1, \cdots, u_n\}$ and $\mathcal{F} = \{S_1, \cdots, S_m\}$ is a family of subsets of U. Without loss of generality, assume that every u_i is in at least one set S_j. Let G be the graph with vertex set $U \cup V \cup \{x\}$ where $U = \{u_1, \cdots, u_n\}$ and $V = \{s_1, \cdots, s_m\}$. The set V is an independent set and the set U induces the path $u_1, \cdots, u_n$ (in order). Further, a vertex u_i is adjacent to s_j if and only if $u_i \in S_j$ and x is adjacent to every vertex in V and to u_1 in U. Clearly, $S = U \cup \{x\}$ is a connected vertex cover of G.

Let G' be the graph obtained from G by deleting the edges with both endpoints in U and the edge xu_1. We claim that $(U, \mathcal{F}, \ell)$ is a yes-instance of Set Cover if and only if $(G, G', S = U \cup \{x\}, k = n, r = \ell)$ is a yes-instance of Dynamic Connected Vertex Cover. Suppose $\mathcal{F}'$ is a set cover of size at most ℓ. Then, $S' = S \cup \{s_i \mid S_i \in \mathcal{F}'\}$ is a connected vertex cover of G' with $d_v(S, S') \leq \ell$. Conversely, suppose G' has a connected vertex cover S' with $d_v(S, S') \leq \ell$. From Observation 9, assume that $S \subseteq S'$ and so $|S' \setminus S| \leq \ell$. Further, $S' \setminus S \subseteq V$. Now, $\{S_i \in \mathcal{F} \mid v_i \in S' \cap V\}$ is a set cover of size at most ℓ. This leads to the claimed lower bound under the Set Cover Conjecture. □

Note that the above reduction also shows that Dynamic Connected Vertex Cover does not have a polynomial kernel when parameterized by k.

Theorem 12. Dynamic Connected Vertex Cover *does not admit a polynomial kernel when parameterized by* k *unless* $\mathsf{NP} \subseteq \mathsf{coNP}/poly$*.*

5 Dynamic Feedback Vertex Set

A *feedback vertex set* is a set of vertices whose deletion results in a forest and Dynamic Feedback Vertex Set is defined as follows.

> Dynamic Feedback Vertex Set
> **Input:** Graphs G, G' on the same vertex set, a feedback vertex set X of G and integers k, r such that $d_e(G, G') \leq k$.
> **Question:** Does there exist a feedback vertex set X' of G' such that $d_v(X, X') \leq r$?

Clearly, Dynamic Feedback Vertex Set is Dynamic Π-Deletion where Π is the set of all forests. As Feedback Vertex Set, the problem of determining if a graph has a feedback vertex set of at most l vertices, is NP-hard, its dynamic variant is NP-hard too by Theorem 1. Feedback Vertex Set is known to admit an $\mathcal{O}^*(3.592^l)$ algorithm [KP14] and a kernel with $\mathcal{O}(l^2)$ vertices [Tho10]. Also, a randomized algorithm that solves the problem in $\mathcal{O}^*(3^l)$ time is known from [CNP+11]. By Theorem 4 and Corollaries 5 and 6, all these results extend to Dynamic Feedback Vertex Set. In particular, the following results hold.

- Dynamic Feedback Vertex Set can be solved in $\mathcal{O}^*(3.592^r)$ time and in $\mathcal{O}^*(3.592^k)$ time.
- Dynamic Feedback Vertex Set admits randomized algorithms with $\mathcal{O}^*(3^r)$ and $\mathcal{O}^*(3^k)$ running times.
- Dynamic Feedback Vertex Set admits an $O(r^2)$ kernel and an $O(k^2)$ kernel.

We now improve these bounds by describing a linear kernel and a faster randomized algorithm for the problem when parameterized by k. First, we describe the linear kernelization.

Theorem 13. Dynamic Feedback Vertex Set *admits a kernel with at most* $4k$ *vertices and* $3k$ *edges.*

Proof. Consider an instance (G, G', X, k, r) of Dynamic Feedback Vertex Set. Observe that if G' is obtained from G by only deleting edges, then X is feedback vertex set of G' too. Also, edges in $E(G') \setminus E(G)$ that have an endpoint in X do not affect the solution. Moreover, from Observation 3, it suffices to search for a feedback vertex set of G' that contains X. Let H be the subgraph of G' induced by $V(G') \setminus X$. From Theorem 4, we have that (G, G', X, k, r) is a yes-instance of Dynamic Feedback Vertex Set if and only if (H, r) is a yes-instance of Feedback Vertex Set. From Corollary 6, it suffices to output a linear kernel of the instance (H, r).

We show that a linear kernel can be obtained by applying two preprocessing rules well known in the literature [BYGNR94, Tho10]. We primarily exploit the fact that H is obtained by adding at most k edges to a forest. This implies that $|E(H)| \leq |V(H)| + k - 1$. First, we apply the following preprocessing rule on H.

Reduction Rule 5.1. *If there is a vertex* v *of degree at most* 1*, then delete* v*.*

This rule is safe (i.e. it preserves minimum feedback vertex sets) as no minimal feedback vertex set of H contains v. Observe that $|E(H)| \leq |V(H)| + k - 1$ is preserved. Next, when this rule is no longer applicable, we apply the following reduction rule.

Reduction Rule 5.2. *If there is a vertex* v *of degree* 2*, then delete* v *and add an edge between its two neighbours.*

Once again this rule is safe as any minimal feedback vertex set of H can be modified into another minimal feedback vertex set of at most the same size that does not contain v. Further, $|E(H)| \leq |V(H)| + k - 1$ still holds as the number of edges decreases by 1 for every vertex deleted as a result of applying the rule. The following property holds on the resulting graph H'' on which Reduction Rules 5.1 and 5.2 are no longer applicable.

Observation 14. *The minimum degree of* H'' *is at least 3 and* $|E(H'')| \leq |V(H'')| + k - 1$.

This implies that $1.5|V(H'')| \leq |E(H'')| \leq |V(H'')| + k - 1$ and hence $|V(H'')| \leq 2k - 2$, $|E(H'')| \leq 3k - 3$. Further, (H, r) is a yes-instance of Feedback Vertex Set if and only if (H'', r) is a yes-instance of Feedback Vertex Set. Thus, from Corollary 6, the kernelized instance corresponding to (G, G', S, k, r) is $(I_{|V(H'')|}, H'', \emptyset, k = |E(H'')|, r)$. □

Next, we proceed to describe an improved FPT algorithm for the problem when parameterized by k. We first need to state some results known for Feedback Vertex Set. There is a deterministic $\mathcal{O}^*(1.7216^n)$ time algorithm and a randomized $\mathcal{O}^*(1.6667^n)$ time algorithm for finding a minimum feedback vertex set of a graph on n vertices [FGLS16]. The *treewidth* of a graph is a parameter that quantifies the closeness of the graph to a tree (see [CFL+15] for the precise definition). If the treewidth of the input graph is upper bounded by tw, then there is a randomized algorithm that computes a minimum feedback vertex set in $\mathcal{O}^*(3^{tw})$ time [CNP+11]. The following result relates the treewidth of a graph to the number of its vertices and edges.

Lemma 15. [FGSS09] *For any* $\epsilon > 0$*, there exists an integer* n_ϵ *such that for every connected graph* G *on* n *vertices and* m *edges with* $n > n_\epsilon$ *and* $1.5n \leq m \leq 2n$*, the treewidth of* G *is upper bounded by* $\frac{m-n}{3} + \epsilon n$. *Moreover, a tree decomposition of the corresponding width can be constructed in polynomial time.*

This theorem along with the described linear kernelization leads to the following result.

Theorem 16. Dynamic Feedback Vertex Set *has a randomized algorithm with* $\mathcal{O}^*(1.6667^k)$ *running time.*

Proof. Consider an instance (G, G', X, k, r) of Dynamic Feedback Vertex Set. Let (H'', r) be the corresponding instance of Feedback Vertex Set obtained from the linear kernelization of Theorem 13. That is, H'' is a graph (not necessarily simple) on n vertices and m edges such that $m \leq n + k - 1$ and $n \leq 2k - 2$. Further, every vertex of H'' has degree at least 3 and hence $m \geq 1.5n$. If $m > 2n$, then as $m \leq n + k - 1$, we have $n < k - 1$. Then, a minimum feedback vertex set of H'' can be obtained in $\mathcal{O}(1.6667^k)$ using the randomized exact exponential-time algorithm described in [FGLS16]. Otherwise, $1.5n \leq m \leq 2n$. Let ϵ be a constant (to be chosen subsequently). Then, let n_ϵ be the integer obtained from Lemma 15 satisfying the required properties. If $n \leq n_\epsilon$, then a minimum feedback vertex set of H'' can be obtained in constant time as n_ϵ is a constant depending only on ϵ. Otherwise, the treewidth of H'' is at most $t = \frac{m-n}{3} + \epsilon n = \frac{m}{3} + n(\epsilon - \frac{1}{3})$. Then, using the randomized algorithm described in [CNP+11], a minimum feedback vertex set of H'' can be obtained in $\mathcal{O}^*(3^t)$ time. Now, by choosing ϵ to be a sufficiently small constant, t can be made arbitrarily close to $\frac{m-n}{3}$. For instance, if $\epsilon = 10^{-10}$, then t is $.\overline{3}m - .333333333\overline{23}n$. As $\frac{m-n}{3} \leq \frac{n+k-1-n}{3} = \frac{k}{3}$, the algorithm in [CNP+11] runs in $\mathcal{O}^*(1.443^k)$ time. □

6 Dynamic Connected Feedback Vertex Set

A *connected feedback vertex set* is a feedback vertex set that induces a connected subgraph and Dynamic Connected Feedback Vertex Set is defined as follows.

Dynamic Connected Feedback Vertex Set
Input: Graphs G, G' on the same vertex set, a connected feedback vertex set S of G and integers k, r such that $d_e(G, G') \leq k$.
Question: Does there exist a connected feedback vertex S' of G' such that $d_v(S, S') \leq r$?

Note that the generic results of section 2 (in particular Corollary 5) do not follow for this problem straightaway. In fact, even a result analogous to Observation 9 does not necessarily hold. We first show that the problem is W[2]-hard when parameterized by r. Then, we show that it is FPT when parameterized by k.

Theorem 17. Dynamic Connected Feedback Vertex Set *is* W*[2]-hard when parameterized by* r.

Proof. We prove the claim by showing that Dynamic Connected Vertex Cover reduces to Dynamic Connected Feedback Vertex Set. This reduction is similar to the classical NP-hardness reduction from Vertex Cover to Feedback Vertex Set. Consider an instance (G, G', S, k, r) of Dynamic Connected Vertex Cover. We will construct an instance $(H, H', S, 3k, r)$ of Dynamic Connected Feedback Vertex Set as follows. The graph H is obtained from G by adding a vertex v_e corresponding to every edge $e = xy \in E(G)$ adjacent to x and y. Similarly, the graph H' is obtained from G' by adding a vertex v_e corresponding to every edge $e = xy \in E(G')$ adjacent to x and y. Vertices in $V(H) \setminus V(H')$ are added as isolated vertices to H' and vertices in $V(H') \setminus V(H)$ are added as isolated vertices to H. This addition ensures that $V(H) = V(H')$. Also, if $d_e(G, G') = k'$, then $d_e(H, H') = 3k'$. Then, S' is a connected vertex cover of G' with $d_v(S, S') \leq r$ if and only if S' is a connected feedback vertex set of H' with $d_v(S, S') \leq r$. □

The above reduction also shows that Dynamic Connected Feedback Vertex Set does not have a polynomial kernel when parameterized by k as Dynamic Connected Vertex Cover does not have one.

Theorem 18. Dynamic Connected Feedback Vertex Set *does not admit a polynomial kernel when parameterized by* k *unless* NP $\subseteq$ coNP*/poly.*

Next, we prove the following result on the existence of a *canonical solution* to an instance of Dynamic Connected Feedback Vertex Set.

Lemma 19. *Consider an instance* (G, G', S, k, r) *of* Dynamic Connected Feedback Vertex Set. *Let* $\mathcal{C} = \{C_1, C_2, \ldots, C_l\}$ *be the set of components of* $G'[S]$*. If* S' *is a connected feedback vertex set of* G'*, then there is a connected feedback vertex set* $\tilde{S}$ *of* G' *with* $d_v(S, \tilde{S}) \leq d_v(S, S')$ *such that for each* $C \in \mathcal{C}$*, either* $V(C) \cap \tilde{S} = \emptyset$ *or* $V(C) \subseteq \tilde{S}$.

Proof. Suppose there is a component $C \in \mathcal{C}$ such that $S' \cap V(C) \neq \emptyset$ and $V(C) \not\subseteq S'$. Let S'' be the set $S' \cup V(C)$. As $G'[S']$ and $G'[C]$ are connected and $S' \cap V(C) \neq \emptyset$, it follows that $G'[S'']$ is also connected. Further, S'' is a feedback vertex set of G' since it contains S'. Finally, as $V(C) \subseteq S$, we have $d_v(S, S'') < d_v(S, S')$. Repeating this procedure for each such component C, we get the desired solution $\tilde{S}$. □

Now, we proceed to describe an algorithm for Dynamic Connected Feedback Vertex Set. Let (G, G', S, k, r) be an instance of Dynamic Connected Feedback Vertex Set. Let G' be obtained from G by k_1 edge additions and k_2 edge deletions. Let E' be the set of edges in $E(G') \setminus E(G)$ that have both endpoints in $V(G' - S)$. Let V' be a minimal set of vertices of $V(G' - S)$ that has at least one endpoint of each edge in E'. Then, $G'[S]$ has $l \leq k_2 + 1$ components and the graph $G' - S$ has a feedback vertex set of size at most k_1. The former property is obvious while the latter follows from the fact that $G' - (S \cup V')$ is a forest and $|V'| \leq k_1$ (as $|E'| \leq k_1$).

Let $\mathcal{C} = \{C_1, C_2, \ldots, C_l\}$ be the set of components of $G'[S]$. Armed with Lemma 19, we first guess the components of $G'[S]$ whose vertices are in solution S'. There are at most 2^l choices for this guess. Let $\mathcal{D} = \{D_1, D_2, \ldots, D_\alpha\}$ be the subset of $\mathcal{C}$ whose vertices are in S'. Let $\mathcal{D}'$ denote $\mathcal{C} \setminus \mathcal{D}$. Let $Z = \bigcup_{C \in \mathcal{D}'} V(C)$ and $I = \bigcup_{C \in \mathcal{D}} V(C)$. If $|Z| > r$, we can skip to the next choice of $\mathcal{D}$. Otherwise, the problem reduces to finding a connected feedback vertex set S' of G' such that $I \subseteq S'$, $Z \cap S' = \emptyset$ and $|S' \cap V(G' - S)| \leq r - |Z|$.

Let $\tilde{G}$ be the graph obtained from G' by contracting every edge in $G[Z]$ and every edge in $G[I]$. Let $\tilde{I}$ and $\tilde{Z}$ denote the set of new vertices introduced as a result of this contraction where each vertex in $\tilde{I}$ corresponds to a component in $\mathcal{D}$ and each vertex in $\tilde{Z}$ corresponds to a component in $\mathcal{D}'$. Observe that $|\tilde{I}| = \alpha$ and $|\tilde{Z}| = l - \alpha$. As contracting an edge cannot introduce new cycles [BL11], we have the following proposition.

Proposition 1. G' *has a connected feedback vertex set* S' *with* $|S' \cap V(G' - S)| \leq r - |Z|$ *such that* $I \subseteq S'$ *and* $Z \cap S' = \emptyset$ *if and only if* $\tilde{G}$ *has a connected feedback vertex set* $\tilde{S}$ *of size at most* $r + \alpha - |Z|$ *such that* $\tilde{I} \subseteq \tilde{S}$ *and* $\tilde{Z} \cap \tilde{S} = \emptyset$.

By construction of the graph $\tilde{G}$, $V(\tilde{G}) = \tilde{I} \cup \tilde{Z} \cup V(G' - S)$. Further, the subgraph of $\tilde{G}$ induced by $V(G' - S)$ is a graph that has a feedback vertex set of size at most k_1. Therefore, this subgraph has treewidth upper bounded by $k_1 + 1$. Then, it follows that the treewidth of $\tilde{G}$ is upper bounded by $k + 2$ as $|\tilde{I} \cup \tilde{Z}| = l \leq k_2 + 1$. Also, a tree decomposition of $\tilde{G}$ with this width can be obtained in polynomial time. Now, the task reduces to the following problem where $\tilde{k} = k + 2$ and $\tilde{r} = r + \alpha - |Z|$.

CONSTRAINED CONNECTED FEEDBACK VERTEX SET **Input:** Graph $\tilde{G}$ with tree decomposition of width at most $\tilde{k}$, disjoint subsets $\tilde{I}$ and $\tilde{Z}$ of $V(\tilde{G})$ and an integer $\tilde{r}$. **Question:** Does there exist a connected feedback vertex $\tilde{S}$ of $\tilde{G}$ such that $|\tilde{S}| \leq \tilde{r}$, $\tilde{I} \subseteq \tilde{S}$ and $\tilde{Z} \cap \tilde{S} = \emptyset$?

From [CNP+11], there is a randomized algorithm solving CONSTRAINED CONNECTED FEEDBACK VERTEX SET in $\mathcal{O}^*(4^{\tilde{k}})$ time when $\tilde{Z} = \emptyset$. This algorithm is based on the cut and count technique which can also be adapted to solve the general version of the problem when $\tilde{Z} \neq \emptyset$. Also, there is a deterministic algorithm (rank-based approach) running in $\mathcal{O}^*(2^{\mathcal{O}(tw)})$ time that finds a minimum connected feedback vertex set where tw is an upper bound on the treewidth of the input graph [BCKN15] which can once again be adapted to solve CONSTRAINED CONNECTED FEEDBACK VERTEX SET. Thus, we have the following result.

Theorem 20. DYNAMIC CONNECTED FEEDBACK VERTEX SET *can be solved in* $\mathcal{O}^*(2^{\mathcal{O}(k)})$ *time.*

7 Dynamic Dominating Set

A *dominating set* of a graph G is a set D of vertices such that $D \cap N[v] \neq \emptyset$ for every $v \in V(G)$. A set $S \subseteq V(G)$ is said to dominate another set $T \subseteq V(G)$ if $T \subseteq N[S]$. DYNAMIC DOMINATING SET is formally defined as follows.

DYNAMIC DOMINATING SET **Input:** Graphs G, G', a dominating set D of G and integers k, r such that $d_e(G, G') \leq k$. **Question:** Does there exist a dominating set D' of G' such that $d_v(D, D') \leq r$?

The problem is NP-complete and W[2]-hard when parameterized by r [DEF+14]. Also, it is FPT when parameterized by k but admits no polynomial kernel unless NP $\subseteq$ coNP/poly. We describe a faster algorithm for this parameterization. First, we observe that it suffices to look for a dominating set with a specific property. Consider an instance (G, G', D, k, r) of DYNAMIC DOMINATING SET. As a set that contains a dominating set is also a dominating set, the following observation holds.

Observation 21. *If* D' *is a dominating set of* G'*, then* $D' \cup D$ *is also a dominating set of* G' *with* $d_v(D, D' \cup D) \leq d_v(D, D')$.

Now, we solve DYNAMIC DOMINATING SET by reducing it to an instance of SET COVER.

Theorem 22. DYNAMIC DOMINATING SET *can be solved in* $\mathcal{O}^*(2^k)$ *time.*

Proof. Consider an instance (G, G', D, k, r) of Dynamic Dominating Set. If G' is obtained from G by only adding edges, then D is dominating set of G'. The only kind of edge deletions that could possibly affect the solution are those that have one endpoint in D and the other endpoint in $V(G') \setminus D$. Further, as $d_e(G, G') \leq k$, $|V(G') \setminus N_{G'}[D]| \leq k$. That is, there are at most k vertices in G' that are not dominated by D. Let H be the subgraph of G' induced by $V(G') \setminus D$. Partition $V(H)$ into two sets $C = N_{G'}(D)$ and $B = V(H') \setminus C$ where $|B| \leq k$.

We claim that (G, G', D, k, r) is a yes-instance of Dynamic Dominating Set if and only if there exists a set $P \subseteq V(H)$ of cardinality at most r such that $B \subseteq N_H[P]$. If there is a set P of size at most r in $V(H)$ that dominates B, then $D' = D \cup P$ is a dominating set of G' with $d_v(D, D') \leq r$. Hence, (G, G', D, k, r) is a yes-instance of Dynamic Dominating Set. Conversely, suppose there is a dominating set D' of G' with $d_v(D, D') \leq r$. Define D'' as $D' \setminus D$. Notice that $|D''| \leq r$. By construction of H, B is not dominated by D and hence $B \subseteq N_H[D'']$. This implies that D'' is the required set of vertices of H that dominates B.

The problem now reduces to finding a set of at most r vertices from $B \cup C$ that dominates B in H. We construct an instance of Set Cover with $U = B$, $\mathcal{F} = \{N_H(u) \cap B \mid u \in C\} \cup \{N_H[w] \cap B \mid w \in B\}$ and $\ell = r$. Then, there exists a set P of size at most r in H which dominates B if and only if $(U, \mathcal{F}, \ell)$ is a yes-instance of Set Cover. A set $X \in \mathcal{F}$ is said to be associated with a vertex v in C if $X = N_H(v) \cap B$ or with a vertex v in B if $X = N_H[v] \cap B$. If there exists a set P with desired property, then every vertex w in B is contained in open or closed neighbourhood of some vertex in P. Consider the subfamily $\mathcal{F}'$ of $\mathcal{F}$ that are associated with vertices in P. Every element of U is contained in at least one of these sets. Thus, $\mathcal{F}'$ is the required set cover. Conversely, if there exists a set cover $\mathcal{F}'$ of size at most $\ell = r$, then let P' be the set of vertices which are associated with sets in $\mathcal{F}'$. Then, $|P'| = |\mathcal{F}'| \leq r$ and every vertex in B is either in P' or is adjacent to some vertex in P'. Hence, P' is the desired set.

As any instance $(U, \mathcal{F}, \ell)$ of Set Cover can be solved in $\mathcal{O}^*(2^{|U|})$ [FKW05], the claimed running time bound follows. □

Finally, we show a lower bound on the running time of an algorithm that solves Dynamic Dominating Set.

Theorem 23. Dynamic Dominating Set *does not admit an algorithm with* $\mathcal{O}^*((2-\epsilon)^k)$ *running time for any* $\epsilon > 0$ *assuming the Set Cover Conjecture.*

Proof. Consider an instance $(U, \mathcal{F}, \ell)$ of Set Cover where $U = \{u_1, \cdots, u_n\}$ and $\mathcal{F} = \{S_1, \cdots, S_m\}$ is a family of subsets of U. Without loss of generality, assume that every u_i is in at least one set S_j. Let G be the graph with vertex set $U \cup V \cup \{x\}$ where $U = \{u_1, \cdots, u_n\}$ and $V = \{s_1, \cdots, s_m\}$. The set V is a clique and the set U is an independent set in G. Further, a vertex u_i is adjacent to s_j if and only if $u_i \in S_j$ and x is adjacent to every vertex in $U \cup V$. Clearly, $D = \{x\}$ is a dominating set of G. Let G' be the graph obtained from G by deleting edges between x and U. We claim that $(U, \mathcal{F}, \ell)$ is a yes-instance of Set Cover if and only if $(G, G', D = \{x\}, k = n, r = \ell)$ is a yes-instance of Dynamic Dominating Set. Suppose $\mathcal{F}'$ is a set cover of size at most ℓ. Then, $D' = D \cup \{s_i \mid S_i \in \mathcal{F}'\}$ is a dominating set of G' with $d_v(D, D') \leq \ell$. Conversely, suppose G' has a dominating set D' with $d_v(D, D') \leq \ell$. From Observation 21, assume that $D \subseteq D'$ and so $|D' \setminus D| \leq \ell$. For every vertex $u \in U \cap D'$, replace u by one of its neighbours in V. The resultant dominating set D'' contains D and satisfies $D'' \setminus D \subseteq V$. Now, $\{S_i \in \mathcal{F} \mid v_i \in D'' \cap V\}$ is a set cover of size at most ℓ. This leads to the claimed lower bound under the Set Cover Conjecture. □

The above reduction also shows that Dynamic Dominating Set does not have a polynomial kernel when parameterized by k.

Theorem 24. Dynamic Dominating Set *does not admit a polynomial kernel when parameterized by* k *unless* $\mathsf{NP} \subseteq \mathsf{coNP}/poly$.

8 Dynamic Connected Dominating Set

A *connected dominating set* is a dominating set that induces a connected graph and Dynamic Connected Dominating Set is formally defined as follows.

> Dynamic Connected Dominating Set
> **Input:** Graphs G, G' on the same vertex set, a connected dominating set D of G and integers k, r such that $d_e(G, G') \leq k$.
> **Question:** Does there exist a connected dominating set D' of G' such that $d_v(D, D') \leq r$?

The problem is NP-complete and admits an $\mathcal{O}^*(4^k)$ time algorithm by a reduction to finding a minimum weight Steiner tree [AKEF+15]. We now show that it has an $\mathcal{O}^*(2^k)$ time algorithm by a reduction to finding a group Steiner tree. Analogous to the problems considered earlier, we first observe a property of the required solution. Consider an instance (G, G', D, k, r) of Dynamic Connected Dominating Set. Observe that G' must be connected, otherwise, (G, G', D, k, r) is a no-instance. As a set that contains a dominating set is also a dominating set and every vertex in $D \setminus D'$ is adjacent to some vertex in D', the following claim holds.

Observation 25. *If D' is a connected dominating set of G', then $D' \cup D$ is also a connected dominating set of G' with $d_v(D, D' \cup D) \leq d_v(D, D')$.*

Now, we describe an algorithm for Dynamic Connected Dominating Set.

Theorem 26. Dynamic Connected Dominating Set *can be solved in $\mathcal{O}^*(2^k)$ time.*

Proof. Consider an instance (G, G', D, k, r) of Dynamic Connected Dominating Set. Observe that G' must be connected, otherwise, it is a no-instance. Also, edges in $E(G') \setminus E(G)$ do not affect the solution. Partition $V(G') \setminus D$ into two sets $C = N_{G'}(D)$ and $B = V(G') \setminus C$. Contract each component of $G'[D]$ to a single vertex. Let H denote the resulting graph and let $X = V(H) \setminus V(G')$. Construct an instance $(H, |X| + r, \mathcal{F})$ of Group Steiner Tree where $\mathcal{F} = \{N_{G'}[v] \mid v \in B\} \cup \{\{x\} \mid x \in X\}$. We claim that (G, G', D, k, r) is a yes-instance of Dynamic Connected Dominating Set if and only if $(H, |X| + r, \mathcal{F})$ is a yes-instance of Group Steiner Tree.

Suppose there exists a connected dominating set D' of G' such that $d_v(D, D') \leq r$ and $D \subseteq D'$. For every vertex u in B, there is a vertex x in $D' \cap (B \cup C)$ that is adjacent to u. As $G'[D']$ is connected, it follows that $H[X \cup (D' \cap (B \cup C))]$ is also connected. Moreover, as $|D' \cap (C \cup B)| \leq |D'| - |D| \leq r$, it follows that the spanning tree of $H[X \cup (D' \cap (C \cup B))]$ is of size at most $|X| + r$. Hence $(H, |X| + r, \mathcal{F})$ is a yes-instance of Group Steiner Tree. Suppose $(H, |X| + r, \mathcal{F})$ is a yes-instance of Group Steiner Tree. Let T denote the solution tree of H. Then, $X \subseteq V(T)$ and $|V(T) \setminus X| = |V(T) \cap (C \cup B)| \leq r$. Define $D' = D \cup (V(T) \cap (B \cup C))$. The size of D' is at most $|D| + r$. Now, $G'[D']$ is connected as D' is obtained from the vertices of T. Also, for every vertex u in B, T contains at least one vertex in $N_{G'}[v]$. Thus, D' is the desired connected dominating set of G'.

As $E(G) \setminus E(G')$, the sum of the number of components of $G'[D]$ and the size of B is upper bounded by $k + 1$. That is, $|\mathcal{F}| \leq k + 1$ and the Group Steiner Tree algorithm of [MPR+12] runs in $\mathcal{O}^*(2^k)$ time. □

Finally, by a reduction from Set Cover to Dynamic Connected Dominating Set, we show the following result.

Theorem 27. Dynamic Connected Dominating Set *does not admit an algorithm with $\mathcal{O}^*((2-\epsilon)^k)$ running time for any $\epsilon > 0$ assuming the Set Cover Conjecture.*

Proof. We observe that the reduction described in Theorem 23 produces instances of Dynamic Connected Dominating Set. Thus, the claimed lower bound holds assuming the Set Cover Conjecture. □

The above reduction also shows that Dynamic Connected Dominating Set does not have a polynomial kernel when parameterized by k.

Theorem 28. Dynamic Connected Dominating Set *does not admit a polynomial kernel when parameterized by* k *unless* $\mathsf{NP} \subseteq \mathsf{coNP}/poly$.

9 Conclusion

We studied the parameterized complexity of several classical problems in the dynamic framework. Recently, the connection between dynamic problems and the improvement of greedy heuristics has been witnessed to a significant extent [HN13, DEF+14, AKCE+17]. In particular, an algorithm for Dynamic Dominating Set has been used to improve heuristic search algorithms for Dominating Set [DEF+14, AKCE+17]. We believe that the heuristics can be improved further using our faster algorithm for Dynamic Dominating Set. Along similar lines, studying the complexity of the dynamic version of various fixed-parameter intractable problems like Partial Vertex Cover, Set Cover and Hitting Set is an exciting direction of research. This could provide an insight into the possibility of (exact or heuristic) algorithms for these problems that are better than the ones currently known. The role of structural parameters like treewidth and pathwidth in the dynamic setting also remains to be explored.

Acknowledgments: We are grateful to Saket Saurabh for the invaluable discussions and for providing several useful pointers.

References

[AKCE+17] F. N. Abu-Khzam, S. Cai, J. Egan, P. Shaw, and K. Wang. Turbo-Charging Dominating Set with an FPT Subroutine: Further Improvements and Experimental Analysis. In *Proceedings of the* 14th *International Conference on Theory and Applications of Models of Computation*, pages 59–70. Springer, 2017.

[AKEF+15] F. N. Abu-Khzam, J. Egan, M. R. Fellows, F. A. Rosamond, and P. Shaw. On the Parameterized Complexity of Dynamic Problems. *Theoretical Computer Science*, 607 (3):426–434, 2015.

[BCKN15] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic Single Exponential Time Algorithms for Connectivity Problems Parameterized by Treewidth. *Information and Computation*, 243:86–111, 2015.

[Bei99] R. Beigel. Finding Maximum Independent Sets in Sparse and General Graphs. In *Proceedings of the* 10th *Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '99, pages 856–857. SIAM, 1999.

[BL11] P. Bonsma and D. Lokshtanov. Feedback Vertex Set in Mixed Graphs. In *Proceedings of the* 12th *International Conference on Algorithms and Data Structures*, WADS '11, pages 122–133. Springer-Verlag, 2011.

[BYGNR94] R. Bar-Yehuda, D. Geiger, J. Naor, and R. M. Roth. Approximation Algorithms for the Vertex Feedback Set Problem with Applications to Constraint Satisfaction and Bayesian Inference. In *Proceedings of the* 5th *Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 344–354. SIAM, 1994.

[Cai96] L. Cai. Fixed-Parameter Tractability of Graph Modification Problems for Hereditary Properties. *Information Processing Letters*, 58(4):171–176, 1996.

[CDL+16] M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlstrom. On Problems As Hard As CNF-SAT. *ACM Transactions on Algorithms*, 12(3):41:1–41:24, 2016.

[CFL+15] M. Cygan, F. V. Fomin, K. Lukasz, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms.* Springer, 2015.

[CKJ01] J. Chen, I. A. Kanj, and W. Jia. Vertex Cover: Further Observations and Further Improvements. *Journal of Algorithms*, 41(2):280–301, 2001.

[CKX10] J. Chen, I. A. Kanj, and G. Xia. Improved Upper Bounds for Vertex Cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010.

[CNP+11] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time. In *Proceedings of the* 52nd *Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 150–159. IEEE, 2011.

[DEF+14] R.G. Downey, J. Egan, M.R. Fellows, F.A. Rosamond, and P. Shaw. Dynamic Dominating Set and Turbo-Charging Greedy Heuristics. *Tsinghua Science and Technology*, 19(4):329–337, 2014.

[DF13] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Springer-Verlag, 2013.

[Die05] R. Diestel. *Graph Theory.* Springer-Verlag, 2005.

[DLS09] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through Colors and IDs. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP 2009)*, pages 378–389. Springer, 2009.

[FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, 2006.

[FGLS16] F. V. Fomin, S. Gaspers, D. Lokshtanov, and S. Saurabh. Exact Algorithms via Monotone Local Search. In *Proceedings of the* 48th *Annual ACM Symposium on Theory of Computing*, STOC '16, pages 764–775. ACM, 2016.

[FGSS09] F. V. Fomin, S. Gaspers, S. Saurabh, and A. A. Stepanov. On Two Techniques of Combining Branching and Treewidth. *Algorithmica*, 54(2):181–207, 2009.

[FKW05] F. V. Fomin, D. Kratsch, and G. J. Woeginger. Exact (Exponential) Algorithms for the Dominating Set Problem. In *Proceedings of the* 30th *Workshop on Graph Theoretic Concepts in Computer Science (WG)*, pages 245–256. Springer, 2005.

[GGJ+17] S. Gaspers, J. Gudmundsson, M. Jones, J. Mestre, and S. Rümmele. Turbocharging Treewidth Heuristics. In *Proceedings of the* 11th *International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017.

[HN13] S. Hartung and R. Niedermeier. Incremental List Coloring of Graphs, Parameterized by Conservation. *Theoretical Computer Science*, 494:86–98, 2013.

[KP14] T. Kociumaka and M. Pilipczuk. Faster Deterministic Feedback Vertex Set. *Information Processing Letters*, 114(10):556–560, 2014.

[KR02] S. Khot and V. Raman. Parameterized Complexity of Finding Subgraphs with Hereditary Properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.

[LY80] J. M. 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.

[MPR+12] N. Misra, G. Philip, V. Raman, S. Saurabh, and S. Sikdar. FPT Algorithms for Connected Feedback Vertex Set. *Journal of Combinatorial Optimization*, 24(2):131–146, 2012.

[Tho10] S. Thomassé. A $4k^2$ Kernel for Feedback Vertex Set. *ACM Transactions on Algorithms*, 6(2):32:1–32:8, 2010.