Tree Deletion Set has a Polynomial Kernel (but no OPT$^{\mathcal{O}(1)}$ approximation)

Archontia C. Giannopoulou[1], Daniel Lokshtanov[1], Saket Saurabh[1,2], and Ondřej Suchý[3]

1 **Department of Informatics, University of Bergen, Norway.**
`{archontia.giannopoulou|daniello}@ii.uib.no`

2 **The Institute of Mathematical Sciences, Chennai, India.**
`saket@imsc.res.in`

3 **Faculty of Information Technology, Czech Technical University Prague, Czech Republic.**
`ondrej.suchy@fit.cvut.cz`

Abstract

In the Tree Deletion Set problem the input is a graph G together with an integer k. The objective is to determine whether there exists a set S of at most k vertices such that $G \setminus S$ is a tree. The problem is NP-complete and even NP-hard to approximate within any factor of OPT^c for any constant c. In this paper we give an $\mathcal{O}(k^5)$ size kernel for the Tree Deletion Set problem. An appealing feature of our kernelization algorithm is a new reduction rule, based on system of linear equations, that we use to handle the instances on which Tree Deletion Set is hard to approximate.

1998 ACM Subject Classification F.2.2, G.2.2

Keywords and phrases Tree Deletion Set, Feedback Vertex Set, Kernelization, Linear Equations

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

In the Tree Deletion Set problem we are given as input an undirected graph G and integer k, and the task is to determine whether there exists a set $S \subseteq V(G)$ of size at most k such that $G \setminus S$ is a tree, that is a connected acyclic graph. This problem was first mentioned by Yannakakis [24] and is related to the classical Feedback Vertex Set problem. Here input is a graph G and integer k and the goal is to decide whether there exists a set S on at most k vertices such that $G \setminus S$ is acyclic. The only difference between the two problems is that in Tree Deletion Set $G \setminus S$ is required to be connected, while in Feedback Vertex Set it is not. Both problems are known to be NP-complete [11, 24].

Despite the apparent similarity between the two problems their computational complexity differ quite dramatically. Feedback Vertex Set admits a factor 2-approximation algorithm, while Tree Deletion Set is known to not admit any approximation algorithm with ratio $\mathcal{O}(n^{1-\epsilon})$ for any $\epsilon > 0$, unless P = NP [1, 24]. With respect to parameterized algorithms, the two problems exhibit more similar behavior. Indeed, some of the techniques that yield fixed parameter tractable algorithms for Feedback Vertex Set [6, 5] can be adapted to also work for Tree Deletion Set [20].

It is also interesting to compare the behavior of the two problems with respect to polynomial time preprocessing procedures. Specifically, we consider the two problems in the realm of *kernelization*. We say that a parameterized graph problem admits a *kernel* of

Conference title on which this volume is based on.
Editors: Billy Editor, Bill Editors; pp. 1–14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

size $f(k)$ if there exists a polynomial time algorithm, called a *kernelization algorithm*, that given as input an instance (G, k) to the problem outputs an equivalent instance (G', k') with $k' \leq f(k)$ and $|V(G')| + |E(G')| \leq f(k)$. If the function f is a polynomial, we say that the problem admits a *polynomial kernel*. We refer to the surveys [12, 17] for an introduction to kernelization. For the Feedback Vertex Set problem, Burrage et al. [4] gave a kernel of size $\mathcal{O}(k^{11})$. Subsequently, Bodlaender [3] gave an improved kernel of size $\mathcal{O}(k^3)$ and finally Thomassé [21] gave a kernel of size $\mathcal{O}(k^2)$. On the other hand the existence of a polynomial kernel for Tree Deletion Set was open until this work. It seems difficult to directly adapt any of the known kernelization algorithms for Feedback Vertex Set to Tree Deletion Set. Indeed, Raman et al. [20] conjectured that Tree Deletion Set does not admit a polynomial kernel.

The main reason to conjecture that Tree Deletion Set does not admit a polynomial kernel stems from an apparent relation between kernelization and approximation algorithms (cf. [18, page 15]). Most problems that admit a polynomial kernel, also have approximation algorithms with approximation ratio polynomial in OPT (cf. [14, page 2]). Here OPT is the value of the optimum solution to the input instance. In fact many kernelization algorithms are already approximation algorithms with approximation ratio polynomial in OPT.

This relation between approximation and kernelization led to a conjecture [19, 9] that Vertex Cover does not admit a kernel with $(2 - \epsilon)k$ vertices for $\epsilon > 0$, as this probably would yield a factor $(2 - \epsilon)$ approximation for the problem thus violating the Unique Games Conjecture [13].

It is easy to show that an approximation algorithm for Tree Deletion Set with ratio $\text{OPT}^{\mathcal{O}(1)}$ would yield an approximation algorithm for the problem with ratio $\mathcal{O}(n^{1-\epsilon})$ thereby proving $\mathtt{P} = \mathtt{NP}$. In particular, suppose Tree Deletion Set had an OPT^c algorithm for some constant c. Since the algorithm will never output a set of size more than n, the approximation ratio of the algorithm is upper bounded by $\min(\text{OPT}^c, \frac{n}{\text{OPT}}) \leq n^{1-\frac{1}{c+1}}$. This rules out approximation algorithms for Tree Deletion Set with ratio $\text{OPT}^{\mathcal{O}(1)}$, and makes it very tempting to conjecture that Tree Deletion Set does not admit a polynomial kernel.

In this paper we show that Tree Deletion Set admits a kernel of size $\mathcal{O}(k^5)$. To the best of our knowledge this is among the few examples of problems that do admit a polynomial kernel, but do not admit any approximation algorithm with ratio $\text{OPT}^{\mathcal{O}(1)}$ under plausible complexity assumptions.

Our methods. The starting point of our kernel are known reduction rules for Feedback Vertex Set adapted to our setting. We also adapt the strategy to model some "pendant parts" of the graph by weight on vertices during the kernelization process to simplify the structure of the graph. By applying these graph theoretical reduction rules we can show that there is a polynomial time algorithm that given an instance (G, k) of Tree Deletion Set outputs an equivalent instance (G', k') and a partition of $V(G')$ into sets B, T, and I such that

1. $|B| = \mathcal{O}(k^2)$,
2. $|T| = \mathcal{O}(k^4)$,
3. I is an independent set, and
4. for every $v \in I$, $N_{G'}(v) \subseteq B$, and $N_{G'}(v)$ is a double clique.

Here a "double clique" means that for every pair x, y of vertices in $N_{G'}(v)$, there are two edges between them. Thus we will allow G' to be a multigraph, and consider a double edge between two vertices as a cycle. In order to obtain a polynomial kernel for Tree Deletion Set it is sufficient to reduce the set I to size polynomial in k.

For every vertex $v \in I$ and tree deletion set S we know that $|N_{G'}(v) \setminus S| \leq 1$, since otherwise $G' \setminus S$ would contain a double edge. Further, if $v \notin S$ then v has to be connected to the rest of $G' \setminus S$ and hence $|N_{G'}(v) \setminus S| = 1$, implying that v is a leaf in $G' \setminus S$. Therefore $G' \setminus (S \cup I)$ must be a tree. We can now reformulate the problem as follows.

For each vertex u in $G' \setminus I$ we have a variable x_u which is set to 0 if $u \in S$ and $x_u = 1$ if $u \notin S$. For each vertex $v \in I$ we have a linear equation $\sum_{u \in N(v)} x_u = 1$. The task is to determine whether it is possible to set the variables to 0 or 1 such that (a) the subgraph of G' induced by the vertices with variables set to 1 is a tree and (b) the number of variables set to 0 plus the number of unsatisfied linear equations is at most k.

At this point it looks difficult to reduce I by graph theoretic means, as performing operations on these vertices correspond to making changes in a system of linear equations. In order to reduce I we prove that there exists an algorithm that given a set $\mathcal{S}$ of linear equations on n variables and an integer k in time $\mathcal{O}(|\mathcal{S}|n^{\omega-1}k)$ outputs a set $\mathcal{S}' \subseteq \mathcal{S}$ of at most $(n+1)(k+1)$ linear equations such that any assignment of the variables that violates at most k linear equations of $\mathcal{S}'$ satisfies all the linear equations of $\mathcal{S} \setminus \mathcal{S}'$. To reduce I we simply apply this result and keep only the vertices of I that correspond to linear equations in $\mathcal{S}'$. We believe that our reduction rule for linear equations will find more applications in the future and, while not as involved, adds a little to the toolbox of algebraic reduction rules for kernelization (see, for example, [15, 16, 22, 7, 8]).

2 Basic Notions

For every positive integer n we denote by $[n]$ the set $\{1, 2, \ldots, n\}$ and for every set S we denote by $\binom{S}{2}$ the 2-subsets of S. $\mathbb{N}$ denotes the natural numbers and $\mathbb{R}$ denotes the real numbers.

For a graph $G = (V, E)$, we use $V(G)$ to denote its vertex set V and $E(G)$ to denote its edge set E. If $S \subseteq V(G)$ we denote by $G \setminus S$ the graph obtained from G after removing the vertices of S. In the case where $S = \{u\}$, we abuse notation and write $G \setminus u$ instead of $G \setminus \{u\}$. For $S \subseteq V(G)$, the *neighborhood* of S in G, $N_G(S)$, is the set $\{u \in G \setminus S \mid \exists v \in S : \{u, v\} \in E(G)\}$. Again, in the case where $S = \{v\}$ we abuse notation and write $N_G(v)$ instead of $N_G(\{v\})$. We use $\mathcal{C}(G)$ to denote the set of the connected components of G. Given a graph class $\mathcal{F}$ we denote by $\mathbf{U}\mathcal{F}$ the set $\cup_{G \in \mathcal{F}} V(G)$. Given two vectors $x, y \in \mathbb{R}^n$, we denote by $\mathbf{d}_H(x, y)$ the Hamming distance of x and y, that is, $\mathbf{d}_H(x, y)$ is equal to the number of positions where the vectors differ. For every $k \in \mathbb{N}$ we denote by $\mathbf{0}^k$ the vector $(0, 0, \ldots, 0) \in \mathbb{R}^k$. When k is implied from the context we abuse notation and denote $\mathbf{0}^k$ as $\mathbf{0}$.

Given a graph G and a set $S \subseteq G$, we say that S is a *feedback vertex set* of G if the graph $G \setminus S$ does not contain any cycles. In the case where $G \setminus S$ is connected we call S *tree deletion set* of G. Moreover, given a set $S \subseteq V(G)$, we say that S is a double clique of G if every pair of vertices in S is joined by a double edge.

For a rooted tree T and vertex set M in $V(T)$ the least common ancestor-closure (*LCA-closure*) **LCA-closure**(M) is obtained by the following process. Initially, set $M' = M$. Then, as long as there are vertices x and y in M' whose least common ancestor w is not in M', add w to M'. Finally, output M' as the LCA-closure of M.

▶ **Lemma 1** ([10]). *Let T be a tree, $M \subseteq V(T)$, and $M' =$ **LCA-closure**(M). Then, $|M'| \leq 2|M|$ and for every connected component C of $T \setminus M'$, $|N(C)| \leq 2$.*

3 A polynomial kernel for Tree Deletion Set

In this section we prove a polynomial size kernel for a weighted variant of the Tree Deletion Set problem. More precisely the problem we will study is following.

Weighted Tree Deletion Set (wTDS)

Instance: A graph G, a function $w : V(G) \to \mathbb{N}$, and a non-negative integer k.

Parameter: k.

Question: Does there exist a set $S \subseteq V(G)$ such that $\sum_{v \in S} w(v) \leq k$ and $G \setminus S$ is a tree?

3.1 Known Reduction Rules for wTDS

In this subsection we state some already known reduction rules for wTDS that are going to be needed during our proofs.

▶ **Reduction Rule 1** ([20]). *If $k < 0$, then answer NO.*

▶ **Reduction Rule 2** ([20]). *If the input graph is disconnected, then delete all vertices in connected components of weight less than $(\sum_{v \in V} w(v)) - k$ and decrease k by the weight of the deleted vertices.*

▶ **Reduction Rule 3** ([20]). *If v is of degree 1 and u is its only neighbor, then delete v and increase the weight of u by the weight of v.*

▶ **Reduction Rule 4** ([20]). *If $v_0, v_1, \ldots, v_l, v_{l+1}$ is a path in the input graph, such that $l \geq 3$ and $\deg(v_i) = 2$ for every $i \in [l]$, then (a) replace the vertices $v_1, \ldots, v_l$ by two vertices u_1 and u_2 with edges $\{v_0, u_1\}$, $\{u_1, u_2\}$, and $\{u_2, v_{l+1}\}$ and with $w(u_1) = \min\{w(v_i) \mid i \in [l]\}$ and $w(u_2) = \left(\sum_{i=1}^{l} w(v_i)\right) - w(u_1)$. Moreover, if $l \geq 2$ and $w(v_0) > k$ or $w(v_{l+1}) > k$, then apply (a) and then (b) delete u_2 and connect u_1 directly to v_{l+1}.*

Given a vertex x of G, an *x-flower of order k* is a set of cycles pairwise intersecting exactly in x. If G has a x-flower of order $k+1$, then x should be in every tree deletion set of weight at most k as otherwise we would need at least $k+1$ vertices to hit all cycles passing through x. Thus the following reduction rule is safe.

▶ **Reduction Rule 5.** *Let (G, k) be an instance of wTDS. If G has a x-flower of order at least $k+1$, then remove x and decrease the parameter k by the weight of x. The resulting instance is $(G \setminus \{x\}, k - w(x))$.*

▶ **Theorem 2** ([21]). *Let G be a multigraph and x be a vertex of G without a self loop. Then in polynomial time we can either check whether there is a x-flower of order $k+1$ or find a set of vertices $Z \subseteq V(G) \setminus \{x\}$ of size at most $2k$ intersecting every cycle containing x.*

▶ **Reduction Rule 6.** *Let (G, k) be an instance of wTDS. If v is a vertex such that $w(v) > k+1$, then let $w(v) = k+1$.*

An instance (G, k) of wTDS is called *semi-reduced* if none of the Reduction Rules 1 - 6 can be applied.

▶ **Theorem 3** ([1]). *There is an $O(\min\{|E(G)| \log |V|, |V|^2\})$ time algorithm that given a graph G that admits a feedback vertex set of size at most k outputs a feedback vertex set of G of size at most $2k$.*

3.2 A structural decomposition

In this subsection we decompose an instance (G, k) of wTDS to an equivalent instance (G', k) where $V(G')$ is partitioned into three sets C_m, C_g, and I, such that the size of C_m and C_g is polynomial in k and I is an independent set. In particular we obtain the following result.

▶ **Lemma 4.** *There is a polynomial time algorithm that given a semi-reduced instance (G, k) of* wTDS *outputs an equivalent instance (G', k') and a partition of $V(G')$ into sets C_m, C_g, and I such that*

(i) $|C_m| \leq 8k^2 + 2k$,
(ii) C_g *induces a forest and* $|C_g| \leq 160k^4 + 248k^3 + 80k^2 - 16k - 8$,
(iii) I *is an independent set, and*
(iv) *for every* $v \in I$, $N_{G'}(v) \subseteq C_m$ *and* $N_{G'}(v)$ *is a double clique.*

Proof. We divide the proof into three parts: (a) the identification of the set C_m and the proof of (i), (b) the transformation of G to an intermediate graph $\widehat{G}$, the identification of the set C_g, and the proof of (ii), and finally, (c) the transformation of $\widehat{G}$ to G', the identification of I and the proof of (iii) and (iv).

Identification of the set C_m and proof of (i). We begin the proof of the lemma with the identification of the set C_m. First notice that every tree deletion set of G of weight at most k is also a feedback vertex set of G of size at most k in the underlying non-weighted graph. Thus, by applying Theorem 3 we may find in polynomial time a feedback vertex set F of G. If $|F| > 2k$ then output NO. Otherwise,

$$|F| \leq 2k. \tag{1}$$

As the instance (G, k) is semi-reduced, Reduction Rule 5 is not applicable and G does not contain a x-flower of order $k+1$, $x \in F$. Therefore, from Theorem 2, we get that for every $x \in F$ we can find in polynomial time a set $Q^x \subseteq V(G) \setminus \{x\}$ intersecting every cycle that goes through x in G and such that

$$|Q^x| \leq 2k, x \in F. \tag{2}$$

Let $Q = \bigcup_{x \in F} Q^x$ and notice that from the definition of Q, and Eq. (1) and (2),

$$|Q| \leq \left| \bigcup_{x\in F} Q^x \right| \leq \sum_{x\in F} |Q^x| \leq \sum_{x \in F} 2k \leq 4k^2. \tag{3}$$

Let $\mathcal{C}(G \setminus F) = \{H_1, H_2, \ldots, H_l\}$ and note that, as F is a feedback vertex set of G, all H_i's are trees. From now on, without loss of generality we will assume that H_i is rooted at some vertex $v_i \in V(H_i)$, $i \in [l]$.

Let $Q_i = V(H_i) \cap Q$, $i \in [l]$. In other words, Q_i denotes the set of vertices of H_i that are also vertices of Q, $i \in [l]$. Let also $\widehat{Q}_i = \textbf{LCA-closure}(Q_i)$, that is, let $\widehat{Q}_i$ denote the least common ancestor-closure of the set Q_i in the tree H_i. Let $\widehat{Q} = \bigcup_{i \in [l]} \widehat{Q}_i$ and note that $\widehat{Q} \cap F = \emptyset$. Observe now that from Lemma 1 and Eq. (3) we get that

$$\left|\widehat{Q}\right| = \left| \bigcup_{i\in[l]} \widehat{Q}_i \right| \leq \sum_{i \in [l]} \left|\widehat{Q}_i\right| \leq 2 \sum_{i \in l} |Q_i| \leq 2\,|Q| \leq 8k^2. \tag{4}$$

Finally, we define C_m to be the set $\widehat{Q} \cup F$. From Eq. (1) and (4) it follows that

$$|C_m| \leq 8k^2 + 2k. \tag{5}$$

and we conclude the first part of our proof.

Transformation of G to an intermediate graph $\widehat{G}$, identification of the set C_g, and proof of (ii). We continue our proof by working towards the identification of the set C_g. First notice that a straightforward implication of Lemma 1 is that we may partition $\mathcal{C}(G \setminus C_m)$ into three sets $\mathcal{C}_0(G \setminus C_m)$, $\mathcal{C}_1(G \setminus C_m)$, and $\mathcal{C}_2(G \setminus C_m)$ in such a way that $\mathcal{C}_i(G \setminus C_m)$ contains all the graphs in $\mathcal{C}(G \setminus C_m)$ that have exactly i neighbors in $\widehat{Q}$, $0 \leq i \leq 2$.

Claim 1. For every connected component $H \in \mathcal{C}(G \setminus C_m)$ and every vertex $y \in C_m$, $|N_G(y) \cap V(H)| \leq 1$, that is, every vertex y of F and every vertex y of $\widehat{Q}$ have at most one neighbor in every connected component H of $G \setminus C_m$.

Proof of Claim 1. Let $y \in C_m$ and $H \in \mathcal{C}(G \setminus C_m)$ and assume to the contrary that $|N_G(y) \cap V(H)| \geq 2$. Then, as H is connected, the graph $G[V(H) \cup \{y\}]$ contains a cycle that goes through y. If $y \in F$, we end up to a contradiction to the facts that $G[V(H) \cup \{y\}] \subseteq G \setminus Q^y$ and the set Q^y intersects every cycle that goes through y. If $y \in \widehat{Q}$, it contradicts to the facts that $G[V(H) \cup \{y\}] \subseteq G \setminus F$ (recall that $\widehat{Q} \cap F = \emptyset$) and $G \setminus F$ is acyclic. $\diamond$

Let $\mathcal{P} = \binom{C_m}{2} \setminus \binom{\widehat{Q}}{2}$, that is, $\mathcal{P} = \{\{y, y'\} \subseteq C_m \mid |\{y, y'\} \cap \widehat{Q}| \leq 1\}$. For every pair $\{y, y'\} \in \mathcal{P}$ we let

$$S(y, y') = \{H \in \mathcal{C}_0(G \setminus C_m) \cup \mathcal{C}_1(G \setminus C_m) \mid \{y, y'\} \subseteq N_G(V(H))\} \tag{6}$$

and

$$c(y, y') = |S(y, y')|. \tag{7}$$

In other words, $S(y, y')$ is the set of the connected components in $\mathcal{C}_0(G \setminus C_m) \cup \mathcal{C}_1(G \setminus C_m)$ whose neighborhood in G contains both y and y' and $c(y, y')$ denotes the cardinality of $S(y, y')$. We partition $\mathcal{P}$ into two sets $\mathcal{P}^{\leq k+1}$ and $\mathcal{P}^{\geq k+2}$. In particular

$$\mathcal{P}^{\leq k+1} = \{\{y, y'\} \in \mathcal{P} \mid c(y, y') \leq k + 1\} \tag{8}$$

$$\mathcal{P}^{\geq k+2} = \{\{y, y'\} \in \mathcal{P} \mid c(y, y') \geq k + 2\} \tag{9}$$

Notice that Claim 1 implies that $c(y, y')$ equals the number of vertex-disjoint paths between y and y' whose internal vertices belong to $\mathcal{C}_0(G \setminus C_m) \cup \mathcal{C}_1(G \setminus C_m)$. Let $\{y, y'\} \in \mathcal{P}^{\geq k+2}$. Observe then that if neither y nor y' belong to a tree deletion set of G we need at least $k + 1$ vertices to hit all the cycles of $\pmb{\cup} S(y, y') \bigcup \{y, y'\}$ as otherwise, from the Pigeonhole Principle, for every $S \subseteq \pmb{\cup} S(y, y')$ with $|S| \leq k$ there exist two connected components H_1 and H_2 in $S(y, y')$ such that $(V(H_1) \cup V(H_2)) \cap S = \emptyset$ and thus the graph induced by $V(H_1) \cup V(H_2) \cup \{y, y'\}$ contains a cycle. This implies that (G, k) is a yes instance if and only if at least one of the vertices y and y' is contained in every tree deletion set of G of weight k. Thus, the instance $(\widehat{G}, k)$ obtained from G after adding double edges between the pairs of vertices y and y', $\{y, y'\} \in \mathcal{P}^{\geq k+2}$, is equivalent to (G, k) and $V(\widehat{G}) = V(G)$. This completes the transformation of G to the intermediate graph $\widehat{G}$. We continue now with the identification of the set C_g and the proof of (ii).

We let $\mathcal{S} = \bigcup\{S(\{y, y'\}) \mid \{y, y'\} \in \mathcal{P}^{\leq k+1}\}$ and finally, we define C_g to be the set

$$C_g = \mathbf{\bigcup}\mathcal{S} \bigcup \mathbf{\bigcup}\mathcal{C}_2(G \setminus C_m). \tag{10}$$

From the definition, it follows that C_g induces a forest. We now prove the upper bound on the size of C_g. We first need the following.

Claim 2.[⋆][1] If H is a connected component of $G \setminus C_m$ then $|V(H)| \leq 8k + 8$.

We now prove an upper bound on $|\mathbf{\bigcup}\,\mathcal{S}|$. Notice first that, by definition, $\mathcal{S}$ can be written as the union of the graph classes $\mathcal{S}_1$ and $\mathcal{S}_2$, where

$$\begin{aligned}\mathcal{S}_1 &= \bigcup_{\{x,y\} \in F \cap \mathcal{P}^{\leq k+1}} (H \in \mathcal{C}_0(G \setminus C_m) \cup \mathcal{C}_1(G \setminus C_m) \mid \{x, y\} \subseteq N_G(H)) \\ \mathcal{S}_2 &= \bigcup_{x \in F} \left(\bigcup_{\substack{y \in \widehat{Q} \\ \{x,y\} \in \mathcal{P}^{\leq k+1}}} (H \in \mathcal{C}_0(G \setminus C_m) \cup \mathcal{C}_1(G \setminus C_m) \mid \{x, y\} \subseteq N_G(H)) \right).\end{aligned}$$

Moreover, $\mathbf{\bigcup}\mathcal{S} = \mathbf{\bigcup}\mathcal{S}_1 \cup \mathbf{\bigcup}\mathcal{S}_2$ and thus, it follows that

$$|\mathbf{\bigcup}\,\mathcal{S}| \leq |\mathbf{\bigcup}\,\mathcal{S}_1| + |\mathbf{\bigcup}\,\mathcal{S}_2| \leq |\mathcal{S}_1| \max\{|V(H)| \mid H \in \mathcal{S}_1\} + |\mathcal{S}_2| \max\{|V(H)| \mid H \in \mathcal{S}_2\}. \tag{11}$$

From Claim 2, in order to prove an upper bound on $|\mathbf{\bigcup}\,\mathcal{S}|$ it is enough to prove upper bounds on $|\mathcal{S}_1|$ and $|\mathcal{S}_2|$. Recall first that for every pair $\{x, y\} \in F \cap \mathcal{P}^{\leq k+1}$ there exist at most $k+1$ connected components containing both x and y in their common neighborhood and therefore

$$|\mathcal{S}_1| \leq \binom{|F|}{2}(k+1) \leq \binom{2k}{2}(k+1) = 2k^3 + k^2 - k. \tag{12}$$

For the upper bound on $\mathcal{S}_2$, for every $x \in F$ we partition the set $\widehat{Q}$ into two sets $R_x^{\leq 1}$ and $R_x^{\geq 2}$ in the following way.

$$\begin{aligned}R_x^{\leq 1} &= \{y \in \widehat{Q} \mid \{x, y\} \in \mathcal{P}^{\leq k+1} \text{ and there exists at most one graph} \\ &\quad\ H \in \mathcal{C}_1(G \setminus C_m) \text{ such that } \{x, y\} \subseteq N_G(H)\} \\ R_x^{\geq 2} &= \{y \in \widehat{Q} \mid \{x, y\} \in \mathcal{P}^{\leq k+1} \text{ and there exist at least two distinct graphs} \\ &\quad\ H_1, H_2 \in \mathcal{C}_1(G \setminus C_m) \text{ such that } \{x, y\} \subseteq N_G(H_1) \cap N_G(H_2)\}.\end{aligned}$$

Observe that

$$|\mathcal{S}_2| \leq \sum_{x \in F} \left(|R_x^{\leq 1}| + |R_x^{\geq 2}|(k+1)\right). \tag{13}$$

For every $x \in F$, it trivially holds that

$$|R_x^{\leq 1}| \leq |\widehat{Q}| \leq 8k^2. \tag{14}$$

Moreover, we claim that for every $x \in F$, $|R_x^{\geq 2}| \leq k$. Indeed, assume to the contrary that $|R_x^{\geq 2}| \geq k+1$ for some $x \in$F. Then there exist $k+1$ vertices $y_i \in \widehat{Q}$, $i \in [k+1]$, such

[1] Proofs of claims marked with ⋆ have been moved to appendix.

that for every i there exist two connected components H_1^i and H_2^i in $\mathcal{C}_1(G \setminus C_m)$ such that $\{x, y\} \subseteq N_G(H_1^i) \cap N_G(H_2^i)$. This implies that the graph induced by the vertex x, the vertices y_i, $i \in [k+1]$, and the graphs H_1^i and H_2^i, $i \in [k+1]$, contains a x-flower of order $k+1$ (notice that, as all the graphs belong to $\mathcal{C}_1(G \setminus C_m)$, they are pairwise disjoint). This is a contradiction to the fact that G is semi-reduced. Therefore, for every $x \in F$,

$$|R_x^{\geq 2}| \leq k. \tag{15}$$

From Eq. (13), (14), and (15) we obtain that

$$|\mathcal{S}_2| \leq 18k^3 + 2k^2. \tag{16}$$

Finally, from Eq. (11), (12), (16), and Claim 2 we get that.

$$|\bigcup \mathcal{S}| \leq 160k^4 + 184k^3 + 16k^2 - 8k. \tag{17}$$

We continue by showing that

$$|\bigcup \mathcal{C}_2(G \setminus C_m)\}| \leq 64k^3 + 64k^2 - 8k - 8. \tag{18}$$

Observe that, from Claim 2, in order to find an upper bound on $|\bigcup \mathcal{C}_2(G \setminus C_m)\}|$ it is enough to find an an upper bound on $|\mathcal{C}_2(G \setminus C_m)|$. We do so with the following claim.

Claim 3. [$\star$] $|\mathcal{C}_2(G \setminus C_m)| \leq |\widehat{Q}| - 1 \leq 8k^2 - 1$.

To conclude the proof of the upper bound on $|C_g|$ we notice that Eq. (18) follows from Claims 2 and 3, and the inequality below.

$$\bigcup |\{V(H) \mid H \in \mathcal{C}_2(G \setminus C_m)\}| \quad \leq \quad |\mathcal{C}_2(G \setminus C_m)| \cdot \max_{H \in \mathcal{C}_2(G \setminus C_m)} |V(H)|.$$

Thus, from Eq. (10), (17), and (18), we obtain that $|C_g| \leq 160k^4 + 248k^3 + 80k^2 - 16k - 8$.

Transformation of $\widehat{G}$ to G', identification of I, and proof of (iii) and (iv). We now continue with the construction of the graph G' and the identification of the set I.

In order to construct the graph G' we first need the following claim for the connected components of $\widehat{G} \setminus (C_m \cup C_g)$.

Claim 4. For every connected component $H \in \mathcal{C}(\widehat{G} \setminus (C_m \cup C_g))$, $N_{\widehat{G}}(V(H)) \subseteq C_m$ and $N_{\widehat{G}}(V(H))$ is a double clique.

Proof of Claim 4. Recall that, $\bigcup \mathcal{C}_2(G \setminus C_m) \subseteq C_g$ and that $C_g \subseteq \bigcup \mathcal{C}(G \setminus C_m)$. It follows that $\mathcal{C}(\widehat{G} \setminus (C_m \cup C_g)) \subseteq \mathcal{C}(G \setminus C_m) \setminus \mathcal{C}_2(G \setminus C_m)$. This implies that for every connected component $H \in \widehat{G} \setminus (C_m \cup C_g)$, $N_{\widehat{G}}(H) \subseteq C_m$. Let now y and y' be two vertices in $\mathcal{P}$ that are not joined by a double edge. By construction of the graph $\widehat{G}$, this implies that $\{y, y'\} \in \mathcal{P}^{\leq k+1}$. Moreover, the definitions of the sets $\mathcal{S}$ and C_g, imply that there is no graph $H \in \mathcal{C}(\widehat{G} \setminus (C_m \cup C_g))$ such that $\{y, y'\} \subseteq N_{C_m}(V(H))$. Therefore, for every graph $H \in \mathcal{C}(\widehat{G} \setminus (C_m \cup C_g))$, the graph induced by $N_{\widehat{G}}(V(H))$ is a double clique. ◇

Finally, let G' be the graph obtained from $\widehat{G}$ after contracting every connected component H of $\widehat{G} \setminus (C_m \cup C_g)$ into a single vertex v_H and setting as $w(v_H) = \sum_{v \in V(H)} w(v)$. We define

I to be the set $V(G') \setminus (C_m \cup C_g)$. Then (iii) follows from construction of I. Claim 4 implies that for every vertex $v \in I$, $N_{G'}(v) \subseteq C_m$. This completes the proof of (iv) and concludes the construction of the instance (G', k) and the identification of the sets C_m, C_g, and I.

It remains to prove that the instances (G, k) and (G', k) are equivalent. As it has already been proved that the instances (G, k) and $(\widehat{G}, k)$ are equivalent it is enough to prove that $(\widehat{G}, k)$ and (G', k) are equivalent. Notice that if (G', k) is a yes instance then $(\widehat{G}, k)$ is also a yes instance; for every vertex v in the tree deletion set of weight at most k of G' we consider the vertex v in the tree deletion set of $\widehat{G}$ whenever $v \in C_m \cup C_g$ and the vertices of the connected component that was contracted to v whenever $v \in I$. In order to prove that if $(\widehat{G}, k)$ is a yes instance then (G', k) is also a yes instance we start with the following.

Claim 5. If there exists a tree deletion set S of $\widehat{G}$ of weight at most k then there exists a tree deletion set $\widehat{S}$ of $\widehat{G}$ of weight at most k such that for every $H \in \mathcal{C}(\widehat{G} \setminus (C_m \cup C_g))$, either $V(H) \subseteq \widehat{S}$ or $V(H) \cap \widehat{S} = \emptyset$.

Proof of Claim 5. Recall that for every $H \in \mathcal{C}(\widehat{G} \setminus (C_m \cup C_g))$ it holds that $N_{\widehat{G}}(V(H))$ is a double clique (Claim 4). Therefore, either $N_{\widehat{G}}(V(H)) \subseteq S$ or there exists a unique vertex of $N_{\widehat{G}}(V(H))$ that does not belong to S. Notice that in the case where $N_{\widehat{G}}(V(H)) \subseteq S$, as $N_{\widehat{G}}(V(H))$ is a separator of $\widehat{G}$, it trivially follows that either $V(H) \subseteq S$ or $S \subseteq \widehat{G} \setminus V(H)$ and the claim holds. Let us now assume that there exists a unique vertex w of $N_{\widehat{G}}(V(H))$ that does not belong to S and that $V(H) \cap S \neq \emptyset$. As, from Claim 1, every vertex of $N_{\widehat{G}}(V(H))$ has exactly one neighbor in H it follows that the graph $\widehat{G}[V(H) \cup \{w\}]$ does not contain a cycle. Moreover, w is a cut vertex of $\widehat{G} \setminus S$ and therefore the graph $(\widehat{G} \setminus S) \cup \widehat{G}[V(H) \cup \{w\}]$ is a tree. Thus in the case where $V(H) \cap S \neq \emptyset$ we can remove the vertices of $V(H)$ from S without introducing any cycles to the graph $(\widehat{G} \setminus S) \cup \widehat{G}[V(H) \cup \{w\}]$. Therefore $S \setminus V(H) \subseteq S$ is also a tree deletion set of $\widehat{G}$ and this concludes the proof of the claim. ◇

Let now $\widehat{S}$ be a tree deletion set of $\widehat{G}$ of weight at most k. From Claim 5 we may assume that for every connected component H of $\widehat{G} \setminus (C_m \cup C_g)$ either $V(H) \subseteq \widehat{S}$ of $V(H) \cap \widehat{S} = \emptyset$. Then it is straightforward to see that the vertex set S consisting of the vertices $(C_m \cup C_g) \cap \widehat{S}$ and the vertices of I that correspond to the connected components of $\widehat{G} \setminus (C_m \cup C_g)$ whose vertices belong to $\widehat{S}$ is a tree deletion set of G' of weight equal to the weight of $\widehat{S}$. ◀

3.3 Results on Linear Equations

▶ **Lemma 5.** *For every matrix $M \in \mathbb{R}^{m \times n}$ and positive integer k, there exists a submatrix $M' \in \mathbb{R}^{m' \times n}$ of M, where $m' \leq n(k+1)$, such that for every $x \in \mathbb{R}^n$ with $\boldsymbol{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$, $\boldsymbol{d}_H(M \cdot x^T, \mathbf{0}) = \boldsymbol{d}_H(M' \cdot x^T, \mathbf{0})$. Furthermore, the matrix M' can be computed in time $O(m \cdot n^{\omega-1}k)$, where ω is the matrix multiplication exponent ($\omega < 2.373$ [23]).*

Proof. In order to identify M' we identify $j_0 + 1 \leq k + 1$ (non-empty) submatrices of $M, B_0, B_1, \ldots, B_{j_0}$ that have at most n rows each in the following way: First, let B_0 be a minimal submatrix of M whose rows span all the rows of M, that is, let B_0 be a base of the vector space generated by the rows of M, and let also M_0 be the submatrix obtained from M after removing the rows of B_0. We identify the rest of the matrices inductively as follows: For every $i \in [k]$, if M_{i-1} is not the empty matrix we let B_i be a minimal submatrix of M_{i-1} whose rows span all the rows of M_{i-1} and finally we let M_i be the matrix occurring from M_{i-1} after removing the rows of B_i.

We now define the submatrix M' of M. Let $j_0 \leq k$ be the greatest integer for which M_{j_0-1} is not the empty matrix. Let M' be the matrix consisting of the union of the rows of the (non-empty) matrices B_0 and B_i, $i \in [j_0]$. As the rank of the matrices M, M_i, $i \in [j_0]$, is upper bounded by n, the matrices B_0, B_i, $i \in [j_0]$, have at most n rows each, and therefore M' has at most $n(j_0+1) \leq n(k+1)$ rows. Observe that if $j_0 < k$ then the union of the rows of the non-empty matrices B_0, B_i, $i \in [j_0]$, contains all the rows of M and thus we may assume that $M' = M$ and the lemma trivially holds. Hence, it remains to prove the lemma for the case where $j_0 = k$, and therefore M' consists of the union of the matrices B_0, B_i, $i \in [k]$. As it always holds that

$$\mathbf{d}_H(M \cdot x^T, \mathbf{0}) \geq \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$$

it is enough to prove that for every $x \in \mathbb{R}^n$ for which $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$, $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) \leq \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$. Thus, it is enough to prove that for every row r of the matrix M'' obtained from M after removing the rows of M', it holds that $\mathbf{d}_H(r \cdot x^T, \mathbf{0}) = 0$. Towards this goal let $x \in \mathbb{R}^n$ be a vector such that $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$. From the Pigeonhole Principle there exists an i_0 such that $\mathbf{d}_H(B_{i_0} \cdot x^T, \mathbf{0}) = 0$, that is, if $r_1, r_2, \ldots, r_{|B_{i_0}|}$ are the rows of B_{i_0} then $r_j \cdot x^T = 0$, for every $j \in [|B_{i_0}|]$. Recall however that the row r of M'' is spanned by the rows $r_1, r_2, \ldots, r_{|B_{i_0}|}$ of B_{i_0}. Therefore, there exist $\lambda_j \in \mathbb{R}$, $j \in [|B_{i_0}|]$, such that $r = \sum_{j \in [|B_{i_0}|]} \lambda_j r_j$. It follows that $r \cdot x^T = \sum_{j \in [|B_{i_0}|]} \lambda_j (r_j \cdot x^T) = 0$ and therefore $\mathbf{d}_H(r \cdot x^T, \mathbf{0}) = 0$. This implies that $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) \leq \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$. Finally, for a rectangular matrix of size $d \times r$, $d \geq r$, Bodlaender et al. [2] give an algorithm that computes a minimum weight column basis in time $\mathcal{O}(dr^{\omega-1})$. By running this algorithm $k+1$ times we can find the matrix M' in time $\mathcal{O}(mn^{\omega-1}k)$ and this completes the proof of the lemma. ◀

▶ **Lemma 6.** *There exists an algorithm that given a set $\mathcal{S}$ of linear equations on n variables and an integer k outputs a set $\mathcal{S}' \subseteq \mathcal{S}$ of at most $(n+1)(k+1)$ linear equations such that any assignment of the variables that violates at most k linear equations of $\mathcal{S}'$ satisfies all the linear equations of $\mathcal{S} \setminus \mathcal{S}'$. Moreover, the running time of the algorithm is $\mathcal{O}(|\mathcal{S}|n^{\omega-1}k)$.*

Proof. Let $x_1, x_2, \ldots, x_n$ denote the n variables and α_{ij} denote the coefficient of x_j in the i-th linear equation of S, $i \in [|\mathcal{S}|]$, $j \in [n]$. Let also $\alpha_{i(n+1)}$ denote the constant term of the i-th linear equation of $\mathcal{S}$. In other words, the i-th equation of $\mathcal{S}$ is denoted as $\alpha_{i1}x_1 + \alpha_{i2}x_2 + \cdots + \alpha_{in}x_n + \alpha_{i(n+1)} = 0$. Finally, let M be the matrix where the j-element of the i-th row is α_{ij}, $i \in [|\mathcal{S}|]$, $j \in [n+1]$. From Lemma 5, it follows that for every positive integer k there exists a submatrix M' of M with at most $(n+1)(k+1)$ rows and $n+1$ columns such that for every $x \in \mathbb{R}^{n+1}$ for which $\mathbf{d}_H(M' \cdot x^T, \mathbf{0}) \leq k$, $\mathbf{d}_H(M \cdot x^T, \mathbf{0}) = \mathbf{d}_H(M' \cdot x^T, \mathbf{0})$ and M' can be computed in time $\mathcal{O}(|\mathcal{S}|n^{\omega-1}k)$. Let $\mathcal{S}'$ be the set of linear equations that correspond to the rows of M'. Let then $x_i = \beta_i$, $\beta_i \in \mathbb{R}$, $i \in [n]$, be an assignment that does not satisfy at most k of the equations of $\mathcal{S}'$. This implies that $\mathbf{d}_H(M' \cdot z, \mathbf{0}) \leq k$, where $z = (\beta_1, \beta_2, \ldots, \beta_n, 1)^T$. Again, from Lemma 5, we get that $\mathbf{d}_H(M \cdot z, \mathbf{0}) = \mathbf{d}_H(M' \cdot z, \mathbf{0})$. Thus, the above assignment satisfies all the linear equations of $\mathcal{S} \setminus \mathcal{S}'$. ◀

3.4 The Main Theorem

In this subsection by combining the structural decomposition of Subsection 3.2 and Lemma 6 from Subsection 3.3 we obtain a kernel for wTDS of size $\mathcal{O}(k^4)$.

▶ **Theorem 7.** wTDS *admits a kernel of size $\mathcal{O}(k^4)$ and $\mathcal{O}(k^4 \log k)$ bits.*

Proof. Let (G,k) be an instance of WTDS. Without loss of generality we may assume that it is semi-reduced and that, from Lemma 4, $V(G)$ can be partitioned into three sets C_m, C_g, and I satisfying the conditions of Lemma 4. Note here that as G is semi-reduced, G is connected and therefore every vertex of I has at least one neighbor in C_m. We construct an instance (G',k) of WTDS in the following way. Let $I=\{v_i \mid i \in [|I|]\}$ and $C_m=\{u_j \mid j \in [|C_m|]\}$. We assign a variable x_j to u_j, $j \in [|C_m|]$, and a linear equation l_i to v_i, $i \in [|I|]$, where l_i is the equation $\sum\limits_{j \in [|C_m|]} \alpha_{ij} x_j - 1 = 0$ and $\alpha_{ij}=1$ if $u_j \in N_G(v_i)$ and 0 otherwise. Let $\mathcal{L}=\{l_i \mid i \in [|I|]\}$ and $\mathcal{L}'$ be the subset of $\mathcal{L}$ obtained from Lemma 6. Let also $I'=\{v_p \in I \mid l_p \in \mathcal{L}'\}$ and $G'=G[C_m \cup C_g \cup I']$. We now prove that (G',k) is equivalent to (G,k).

We first prove that if (G,k) is a yes instance then so is (G',k). Let S be a tree deletion set of G of weight at most k. Then $G \setminus S$ is a tree and as for every vertex $v \in I \setminus S$, $G[N_G(v)]$ is a double clique then v has degree exactly 1 in $G \setminus S$. Therefore, the graph obtained from $G \setminus S$ after removing $(I \setminus I')$ is still a tree. This implies that $S \setminus (I \setminus I')$ is a tree deletion set of G' of weight at most k and (G',k) is a yes instance.

Let now (G',k) be a yes instance and S be a tree deletion set of G' of weight at most k. We claim that there exist at most k vertices in I' whose neighborhood lies entirely in S. Indeed, assume to the contrary that there exist at least $k+1$ vertices of I' whose neighborhood lies entirely in S. Let J be the set of those vertices. Notice that for every vertex $v \in I'$, if $N_{G'}(v) \subseteq S$, then either $v \in S$ or $I' \setminus \{v\} \subseteq S$. Notice that if $J \subseteq S$, then S has weight at least $k+1$, a contradiction. Therefore, there exists a vertex $w \in J$ that is not contained in S. Then $I' \setminus \{w\} \subseteq S$. Moreover, recall that w has at least one neighbor z in C_m and from the hypothesis z is contained in S. Therefore $(I' \setminus \{w\}) \cup \{z\} \subseteq S$. As $|I'| \geq |J| = k+1$, it follows that $|I' \setminus \{w\}| \geq k$. Furthermore, recall that $C_m \cap I' = \emptyset$. Thus, $|S| \geq k+1$, a contradiction to the fact that S has weight at most k. Therefore, there exist at most k vertices of I' whose neighborhood is contained entirely in S. For every $j \in [|C_m|]$, let $x_j = \beta_j$, where $\beta_j = 0$ if $u_j \in S$ and 1 otherwise. Then there exist at most k linear equations in $\mathcal{L}'$ which are not satisfied by the above assignment. However, from the choice of $\mathcal{L}'$ all the linear equations in $\mathcal{L} \setminus \mathcal{L}'$ are satisfied and therefore, every vertex in $I \setminus I'$ has exactly one neighbor in $G \setminus S$. Thus $G \setminus S$ is a tree and hence, S is a tree deletion set of G as well.

Notice that $V(G') = C_m \cup C_g \cup I'$, where $|I'| \leq 8k^3 + 10k^2 + 3k + 1$ (Lemma 6) and therefore $|V(G')| = \mathcal{O}(k^4)$. It is also easy to see that $|E(G')| = \mathcal{O}(k^4)$. Indeed, notice first that as the set I' is an independent set there are no edges between its vertices. Moreover, from Lemma 4 there are no edges between the vertices of the set I' and the set C_g. Observe that, from the construction of I and subsequently of I', Lemma 4 implies that every vertex of I' has at most $2k+1$ neighbors in C_m. As $|I'| \leq 8k^3 + 10k^2 + 3k + 1$ there exist $\mathcal{O}(k^4)$ edges between the vertices of I' and the vertices of C_m. Notice that from (ii) of Lemma 4, C_g induces a forest and thus there exist at most $\mathcal{O}(k^4)$ edges between its vertices. Moreover, from (i) of Lemma 4, again there exist $\mathcal{O}(k^4)$ edges between the vertices of C_m. It remains to show that there exist $\mathcal{O}(k^4)$ edges with one endpoint in C_m and one endpoint in C_g. Recall first that every $x \in C_m$ has at most one neighbor in every connected component of $G[C_g]$ and that every connected component has at most 2 neighbors in Q. Therefore, there exist at most $2k+2$ edges between every connected component of $G[C_g]$ and C_m. Moreover, from Eq. (10),(11), (12), and (16), and Claim 3 we obtain that C_g contains $\mathcal{O}(k^3)$ connected components. Therefore, there exist $\mathcal{O}(k^4)$ edges with one endpoint in C_m and one endpoint in C_g. Thus, WTDS has a kernel of $\mathcal{O}(k^4)$ vertices and edges. Finally, from Reduction Rule 6, the weight of every vertex is upper bounded by $k+1$ and thus, it can be encoded using $\log(k+1)$ bits resulting to a kernel of wTDS with $\mathcal{O}(k^4 \log k)$ bits. ◀

References

1 Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discrete Math.*, 12(3):289–297, 1999.

2 Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. In *ICALP (1)*, volume 7965, pages 196–207, 2013.

3 Hans L. Bodlaender and Thomas C. van Dijk. A cubic kernel for feedback vertex set and loop cutset. *Theory Comput. Syst.*, 46(3):566–597, 2010.

4 Kevin Burrage, Vladimir Estivill-Castro, Michael R. Fellows, Michael A. Langston, Shev Mac, and Frances A. Rosamond. The undirected feedback vertex set problem has a poly(k) kernel. In *IWPEC*, volume 4169, pages 192–202, 2006.

5 Yixin Cao, Jianer Chen, and Yang Liu. On feedback vertex set new measure and new structures. In *Algorithm Theory - SWAT 2010*, volume 6139 of *LNCS*, pages 93–104, 2010.

6 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for feedback vertex set problems. *J. Comput. Syst. Sci.*, 74(7):1188–1198, 2008.

7 Robert Crowston, Michael R. Fellows, Gregory Gutin, Mark Jones, Frances A. Rosamond, Stéphan Thomassé, and Anders Yeo. Simultaneously satisfying linear equations over f_2: Maxlin2 and max-r-lin2 parameterized above average. In *FSTTCS*, pages 229–240, 2011.

8 Robert Crowston, Gregory Gutin, Mark Jones, Eun Jung Kim, and Imre Z. Ruzsa. Systems of linear equations over f_2 and problems parameterized above average. In *SWAT*, pages 164–175, 2010.

9 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity.* Texts in Computer Science. Springer, 2013.

10 Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *FOCS*, pages 470–479. IEEE Computer Society, 2012.

11 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

12 Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

13 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

14 Stefan Kratsch. Polynomial kernelizations for MIN $F^+\Pi$ and MAX NP. *Algorithmica*, 63(1-2):532–550, 2012.

15 Stefan Kratsch and Magnus Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In *SODA*, pages 94–103, 2012.

16 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459, 2012.

17 Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Kernelization - preprocessing with a guarantee. In *The Multivariate Algorithmic Revolution and Beyond*, volume 7370, pages 129–161, 2012.

18 Dániel Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.

19 Rolf Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications).* Oxford University Press, USA, March 2006.

20 Venkatesh Raman, Saket Saurabh, and Ondrej Suchý. An FPT algorithm for tree deletion set. In *WALCOM*, volume 7748, pages 286–297, 2013.

21 Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.

22 Magnus Wahlström. Abusing the tutte matrix: An algebraic instance compression for the K-set-cycle problem. In *STACS*, volume 20, pages 341–352, 2013.

23 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In Howard J. Karloff and Toniann Pitassi, editors, *STOC*, pages 887–898. ACM, 2012.

24 Mihalis Yannakakis. The effect of a connectivity requirement on the complexity of maximum subgraph problems. *J. ACM*, 26(4):618–630, 1979.

4 Appendix

4.1 Proof of Claim 2

Let H be a connected component of $G \setminus C_m$. First recall that, from Claim 1, every vertex of C_m has at most 1 neighbor in H and, from construction of $\widehat{Q}$, H has at most 2 neighbors in $\widehat{Q}$. This implies that there are at most $|F| + 2 \leq 2k + 2$ vertices in H that have a neighbor in $G \setminus V(H)$, and in particular in C_m. We call this set of vertices N. Let H_1 be the set of vertices of degree 1 in H, that is, the leaves of H. From Reduction Rule 3 it follows that for every $v \in H_1$, $\deg_G(v) \geq 2$ and thus, as $H \in \mathcal{C}(G \setminus C_m)$, v has at least one neighbor in C_m. Therefore, $H_1 \subseteq N$ and

$$|H_1| \leq 2k + 2.$$

Let now H_3 be the set of vertices of degree at least 3 in H. For H_3 it is easy to observe that, by standard combinatorial arguments on trees,

$$|H_3| \leq |H_1| \leq 2k + 2.$$

Finally, let P be the set $N \cup H_3$ and $\mathcal{E}$ be the set of paths in H with endpoints in P. Again, as $|P| \leq 4k + 4$, it holds that $|\mathcal{E}| \leq 2k + 2$. Observe that by construction of $\mathcal{E}$ all the inner vertices of the paths in $\mathcal{E}$ have degree exactly 2. Therefore, from Reduction Rule 4 we get that every path in $\mathcal{E}$ contains at most 2 vertices. This implies that $|V(H) \setminus P| \leq 4k+4$. To conclude, as $|V(H)| = |V(H) \setminus P| + |P|$, we get that $|V(H)| \leq 8k+8$. $\diamond$

4.2 Proof of Claim 3

Proof of Claim 3. Let $C = \bigcup_{H \in \mathcal{C}_2(G \setminus C_m)} (N_G(H) \cap \widehat{Q})$, that is, let C be the set of neighbors of the graphs in $\mathcal{C}_2(G \setminus C_m)$ in $\widehat{Q}$. Let A_C be the graph with vertex set C where two vertices in C are connected by an edge if and only if they are the neighbors of a graph $H \in \mathcal{C}_2(G \setminus C_m)$ in $\widehat{Q}$. Hence, the number of edges of A_C equals $|\mathcal{C}_2(G \setminus C_m)|$. Notice then that if A_C is a forest the claim follows from Eq. (4). We now work towards showing that A_C is a forest. Indeed, assume to the contrary that there exists a cycle in A_C. Then it is easy to see that we may find a cycle in the graph $\widehat{H}$ induced by the graphs in $\mathcal{C}_2(G \setminus C_m)$ which correspond to the edges of the cycle in A_C and their neighborhood in $\widehat{Q}$. Recall that $\widehat{Q} \cap F = \emptyset$ and therefore $\widehat{H}$ is a subgraph of $G \setminus F$. This contradicts to the fact that F is a feedback vertex set of G and completes the proof of the claim. $\diamond$