Contracting graphs to paths and trees*

Pinar Heggernes[1], Pim van 't Hof[1], Benjamin Lévêque[2], Daniel Lokshtanov[3], and Christophe Paul[2]

[1] Department of Informatics, University of Bergen, N-5020 Bergen, Norway
{pinar.heggernes,pim.vanthof}@ii.uib.no
[2] CNRS, LIRMM, Université Montpellier 2, Montpellier, France
{leveque,paul}@lirmm.fr
[3] Dept. Computer Science and Engineering, University of California San Diego, USA
dlokshtanov@cs.ucsd.edu

Abstract. Vertex deletion and edge deletion problems play a central role in Parameterized Complexity. Examples include classical problems like Feedback Vertex Set, Odd Cycle Transversal, and Chordal Deletion. The study of analogous edge contraction problems has so far been left largely unexplored from a parameterized perspective. We consider two basic problems of this type: Tree Contraction and Path Contraction. These two problems take as input an undirected graph G and an integer k, and the task is to determine whether we can obtain an acyclic graph or a path, respectively, by a sequence of at most k edge contractions in G. We present an algorithm with running time $4.98^k\, n^{O(1)}$ for Tree Contraction, based on a variant of the color coding technique of Alon, Yuster and Zwick, and an algorithm with running time $2^{k+o(k)} + n^{O(1)}$ for Path Contraction. Furthermore, we show that Path Contraction has a kernel with at most $5k + 3$ vertices, while Tree Contraction does not have a polynomial kernel unless NP $\subseteq$ coNP/poly. We find the latter result surprising, because of the connection between Tree Contraction and Feedback Vertex Set, which is known to have a kernel with $O(k^2)$ vertices.

1 Introduction

For a graph class Π, the Π-Contraction problem takes as input a graph G and integer k, and the question is whether there is a graph $H \in \Pi$ such that G can be contracted to H using at most k edge contractions. In early papers by Watanabe et al. [29, 30] and Asano and Hirata [2], Π-Contraction was proved to be NP-complete for several classes Π. The Π-Contraction problem fits into a wider and well studied family of graph modification problems, where vertex deletions and edge deletions are two other ways of modifying a graph. Π-Vertex Deletion and Π-Edge Deletion are the problems of deciding whether some graph belonging to graph class Π can be obtained from G by at most k vertex

* Supported by the Research Council of Norway (project SCOPE, 197548/V30) and the French ANR (project AGAPE, ANR-09-BLAN-0159).

deletions or by at most k edge deletions, respectively. All of these problems are shown to be NP-complete for most of the interesting graph classes Π [25, 31–33]. However, whereas Π-Vertex Deletion and Π-Edge Deletion have been studied in detail for several graph classes Π with respect to fixed parameter tractability (e.g., [3, 5, 10, 17, 19, 21–23, 26, 28]), this has not been the case for Π-Contraction. Very recently, four of the authors proved that Π-Contraction is fixed parameter tractable when Π is the class of bipartite graphs [20], which is, to our knowledge, the only result of this type so far. Note that every edge contraction reduces the number of vertices of the input graph by one, which means that the parameter k of Π-Contraction is never more than $n-1$.

Here we study Π-Contraction when Π is the class of acyclic graphs and when Π is the class of paths. Since edge contractions preserve the number of connected components, we may assume that the input graph is connected, justifying the names Tree Contraction and Path Contraction. Both problems are NP-complete [2, 9]. We find these problems of particular interest, since their vertex deletion versions, widely known as Feedback Vertex Set and Longest Induced Path, are famous and well-studied. These two problems are known to be fixed parameter tractable and have polynomial kernels, when parameterized by the number of deleted vertices.

The question whether a fixed parameter tractable problem has a polynomial kernel or not has attracted considerable attention during the last years, especially after the establishment of methods for proving non-existence of polynomial kernels, up to some complexity theoretical assumptions [6–8]. During the last decade, considerable effort has also been devoted to improving the parameter dependence in the running time of classical parameterized problems. Even in the case of a running time which is single exponential in k, lowering the base of the exponential function is considered to be an important challenge. For instance, the running time of Feedback Vertex Set has been successively improved from $37.7^k\, n^{O(1)}$ [18] to $10.57^k\, n^{O(1)}$ [15], $5^k\, n^{O(1)}$ [12], $3.83^k\, n^{O(1)}$ [11], and randomized $3^k\, n^{O(1)}$ [14].

In this paper, we present results along these established lines for Tree Contraction and Path Contraction. It can be shown that if a graph G is contractible to a path or a tree with at most k edge contractions, then the treewidth of G is $O(k)$. Consequently, when parameterized by k, fixed parameter tractability of Tree Contraction and Path Contraction follows from the well known result of Courcelle [13], as both problems are expressible in monadic second order logic. However, this approach yields very unpractical algorithms whose running times involve huge functions of k. Here, we give algorithms with running time $2^{k+o(k)} + n^{O(1)}$ for Path Contraction, and $4.98^k\, n^{O(1)}$ for Tree Contraction. To obtain the latter result, we use a variant of the color coding technique of Alon, Yuster and Zwick [1]. Combined with the recent results by Cygan et al. [14], our results also imply a randomized algorithm for Tree Contraction with running time $4^{k+o(k)}\, n^{O(1)}$. Furthermore, we show that Path Contraction has a linear vertex kernel. On the negative side, we show that Tree Contraction does not have a polynomial kernel, unless NP

$\subseteq$ coNP/poly. This is a contrast compared to the corresponding vertex deletion version, as FEEDBACK VERTEX SET has a quadratic kernel [27].

2 Definitions and notation

All graphs in this paper are finite, undirected, and simple, i.e., do not contain multiple edges or loops. Given a graph G, we denote its vertex set by $V(G)$ and its edge set by $E(G)$. We also use the ordered pair $(V(G), E(G))$ to represent G. We let $n = |V(G)|$. Let $G = (V, E)$ be a graph. The *neighborhood* of a vertex v in G is the set $N_G(v) = \{w \in V \mid vw \in E\}$ of *neighbors* of v in G. Let $S \subseteq V$. We write $N_G(S)$ to denote $\bigcup_{v\in S} N_G(v) \setminus S$. We say that S *dominates* a set $T \subseteq V$ if every vertex in T either belongs to S or has at least one neighbor in S. We write $G[S]$ to denote the subgraph of G *induced* by S. We use shorthand notation $G-v$ to denote $G[V \setminus \{v\}]$ for a vertex $v \in V$, and $G-S$ to denote $G[V \setminus S]$ for a set of vertices $S \subseteq V$. A graph is *connected* if it has a path between every pair of its vertices, and is *disconnected* otherwise. The *connected components* of a graph are its maximal connected subgraphs. We say that a vertex subset $S \subseteq V$ is *connected* if $G[S]$ is connected. A *bridge* in a connected graph is an edge whose deletion results in a disconnected graph. A *cut vertex* in a connected graph is a vertex whose deletion results in a disconnected graph. A graph is 2-*connected* if it has no cut vertex. A 2-*connected component* of a graph G is a maximal 2-connected subgraph of G.

We use P_ℓ to denote the graph isomorphic to a path on ℓ vertices, i.e., the graph with ordered vertex set $\{p_1, p_2, p_3, \ldots, p_\ell\}$ and edge set $\{p_1p_2,\ p_2p_3, \ldots, p_{\ell-1}p_\ell\}$. We will also write $p_1p_2 \cdots p_\ell$ to denote P_ℓ. A *tree* is a connected acyclic graph. A vertex with exactly one neighbor in a tree is called a *leaf*. A *star* is a tree isomorphic to the graph with vertex set $\{a, v_1, v_2, \ldots, v_s\}$ and edge set $\{av_1, av_2, \ldots, av_s\}$. Vertex a is called the *center* of the star.

The *contraction* of edge xy in G removes vertices x and y from G, and replaces them by a new vertex, which is made adjacent to precisely those vertices that were adjacent to at least one of the vertices x and y. A graph G is *contractible* to a graph H, or H-*contractible*, if H can be obtained from G by a sequence of edge contractions. Equivalently, G is H-contractible if there is a surjection $\varphi : V(G) \to V(H)$, with $W(h) = \{v \in V(G) \mid \varphi(v) = h\}$ for every $h \in V(H)$, that satisfies the following three conditions: **(1)** for every $h \in V(H)$, $W(h)$ is a connected set in G; **(2)** for every pair $h_i, h_j \in V(H)$, there is an edge in G between a vertex of $W(h_i)$ and a vertex of $W(h_j)$ if and only if $h_ih_j \in E(H)$; **(3)** $\mathcal{W} = \{W(h) \mid h \in V(H)\}$ is a partition of $V(G)$. We say that $\mathcal{W}$ is an *H-witness structure* of G, and the sets $W(h)$, for $h \in V(H)$, are called *witness sets* of $\mathcal{W}$.

If a witness set contains more than one vertex of G, then we call it a *big* witness set; a witness set consisting of a single vertex of G is called *small*. We say that G is *k-contractible* to H, with $k \leq n-1$, if H can be obtained from G by at most k edge contractions. The next observation follows from the above.

Observation 1 *If a graph G is k-contractible to a graph H, then $|V(G)| \leq |V(H)| + k$, and any H-witness structure $\mathcal{W}$ of G satisfies the following three* *properties: no witness set of $\mathcal{W}$ contains more than $k+1$ vertices, $\mathcal{W}$ has at most k big witness sets, and all the big witness sets of $\mathcal{W}$ together contain at most $2k$ vertices.*

A 2-*coloring* of a graph G is a function $\phi : V(G) \to \{1, 2\}$. Here, a 2-coloring of G is merely an assignment of colors 1 and 2 to the vertices of G, and should not be confused with a *proper* 2-coloring of G, which is a 2-coloring with the additional property that no two adjacent vertices receive the same color. If all the vertices belonging to a set $S \subseteq V(G)$ have been assigned the same color by ϕ, we say that S is *monochromatic* with respect to ϕ, and we use $\phi(S)$ to denote the color of the vertices of S. Any 2-coloring ϕ of G defines a partition of $V(G)$ into two sets V_ϕ^1 and V_ϕ^2, which are the sets of vertices of G colored 1 and 2 by ϕ, respectively. A set $X \subseteq V(G)$ is a *monochromatic component* of G with respect to ϕ if $G[X]$ is a connected component of $G[V_\phi^1]$ or a connected component of $G[V_\phi^2]$. We say that two different 2-colorings ϕ_1 and ϕ_2 of G *coincide* on a vertex set $A \subseteq V(G)$ if $\phi_1(v) = \phi_2(v)$ for every vertex $v \in A$.

3 Tree Contraction

Asano and Hirata [2] showed that Tree Contraction is NP-complete. In this section, we first show that Tree Contraction does not have a polynomial kernel, unless NP $\subseteq$ coNP/poly. We then present a $4.98^k n^{O(1)}$ time algorithm for Tree Contraction.

A *polynomial parameter transformation* from a parameterized problem Q_1 to a parameterized problem Q_2 is a polynomial time reduction from Q_1 to Q_2 such that the parameter of the output instance is bounded by a polynomial in the parameter of the input instance. Bodlaender et al. [8] proved that if Q_1 is NP-complete, Q_2 is in NP, there is a polynomial parameter transformation from Q_1 to Q_2, and Q_2 has a polynomial kernel, then Q_1 has a polynomial kernel.

Theorem 1. Tree Contraction *does not have a kernel with size polynomial in k, unless NP $\subseteq$ coNP/poly.*

Proof. We give a polynomial parameter transformation from Red-Blue Domination to Tree Contraction. Red-Blue Domination takes as input a bipartite graph $G = (A, B, E)$ and an integer t, and the question is whether there exists a subset of at most t vertices in B that dominates A. We may assume that every vertex of A has a neighbor in B, and that $t \leq |A|$. This problem, when parameterized by $|A|$, has been shown not to have a polynomial kernel, unless NP $\subseteq$ coNP/poly [16]. Since Tree Contraction is in NP, the existence of the polynomial parameter transformation described below implies that Tree Contraction does not have a kernel with size polynomial in k, unless NP $\subseteq$ coNP/poly.

Given an instance of Red-Blue Domination, that is a bipartite graph $G = (A, B, E)$ and an integer t, we construct an instance (G', k) of Tree Contraction with $G' = (A' \cup B', E')$ as follows. To construct G', we first add a new vertex a to A and make it adjacent to every vertex of B. We define $A' = A \cup \{a\}$.

We then add, for every vertex u of A, $k+1$ new vertices to B that are all made adjacent to exactly u and a. The set B' consists of the set B and the $|A|(k+1)$ newly added vertices. Finally, we set $k = |A| + t$. This completes the construction. The rest of the proof is given in the appendix. $\square$

As a contrast to this negative result, we present below an algorithm for Tree Contraction with running time $4.98^k n^{O(1)}$. The proof of the following lemma is given in the appendix.

Lemma 1. *A graph is k-contractible to a tree if and only if each of its 2-connected components can be contracted to a tree, using at most k edge contractions in total.*

The main idea for our algorithm for Tree Contraction is to use 2-colorings of the input graph G. Let T be a tree, and let $\mathcal{W}$ be a T-witness structure of a graph G. We say that a 2-coloring ϕ of G is *compatible* with $\mathcal{W}$ (or $\mathcal{W}$-*compatible*) if the following two conditions are both satisfied: **(1)** every witness set of $\mathcal{W}$ is monochromatic with respect to ϕ, and **(2)** if $W(u)$ and $W(v)$ are big witness sets and $uv \in E(T)$, then $\phi(W(u)) \neq \phi(W(v))$.

Let ϕ be a given 2-coloring of a 2-connected graph G. In Lemma 2, we will show that if ϕ is $\mathcal{W}$-compatible, then we can use the monochromatic components of G with respect to ϕ to compute a T'-witness structure of G, such that T' is a tree with at least as many vertices as T. Informally, we do this by finding a "star-like" partition of each monochromatic component M of G, where one set of the partition is a connected vertex cover of $G[M]$, and all the other sets have size 1. A *connected vertex cover* of a graph G is a subset $V' \subseteq V(G)$ such that $G[V']$ is connected and every edge of G has at least one endpoint in V'.

Proposition 1 **([4]).** *Given a graph G, it can be decided in $2.4882^t n^{O(1)}$ time whether G has a connected vertex cover of size at most t. If such a connected vertex cover exists, then it can be computed within the same time.*

Lemma 2. *Let ϕ be a 2-coloring of a 2-connected graph G. If ϕ is compatible with a T-witness structure of G whose largest witness set has size d, where T is a tree, then a T'-witness structure of G can be computed in time $2.4882^d\, n^{O(1)}$, such that T' is a tree with as at least as many vertices as T.*

Proof. Suppose ϕ is compatible with a T-witness structure $\mathcal{W}$ of G, such that T is a tree, and the largest witness set of $\mathcal{W}$ has size d. The 2-connectedness of G implies that, if a witness set $W(v) \in \mathcal{W}$ is small, then v is a leaf of T.

Let $\mathcal{X}$ be the set of monochromatic components of G with respect to ϕ. Every witness set of $\mathcal{W}$ is monochromatic by property (1) of a $\mathcal{W}$-compatible 2-coloring, and connected by definition. Hence, for every $W \in \mathcal{W}$, there exists an $X \in \mathcal{X}$ such that $W \subseteq X$. Moreover, since every $X \in \mathcal{X}$ is connected, there exists a vertex subset $Y \subseteq V(T)$ such that $T[Y]$ is a connected subtree of T and $X = \bigcup_{y \in Y} W(y)$. Hence, $\mathcal{X}$ is a T''-witness structure of G for a tree T'' that has at most as many vertices as T. We now show how to partition the big witness sets of $\mathcal{X}$ in such a way, that we obtain a T'-witness structure of G for some tree T' with at least as many vertices as T.

Suppose there exists a set $X \in \mathcal{X}$ that contains more than one witness set of $\mathcal{W}$, say $W(v_1), \ldots, W(v_p)$ for some $p \geq 2$. As a result of the observation we made earlier and properties (1) and (2) of a $\mathcal{W}$-compatible 2-coloring, we know that at most one of those sets can be big. If all the sets $W(v_1), \ldots, W(v_p)$ are small, then all the vertices $v_1, \ldots, v_p$ are leaves in T. This means that $p = 2$ and T consists of only two vertices; a trivial case. Suppose one of the sets, say $W(p_1)$, is big. Since each of the sets $W(v_2), \ldots, W(v_p)$ is small, the vertices $v_2, \ldots, v_p$ are leaves in T. This means that the vertices $v_1, \ldots, v_p$ induce a star in T, with center v_1 and leaves $v_2, \ldots, v_p$. Note that $W(v_1)$ is a connected vertex cover in the graph $G[X]$; this observation will be used in the algorithm below. Also note that the sets $W(v_1), \ldots, W(v_p)$ define an S-witness structure $\mathcal{S}$ of the graph $G[X]$, where S is a star with $p - 1$ leaves.

We use the above observations to decide, for each $X \in \mathcal{X}$, if we can partition X into several witness sets. Recall that, given ϕ, we only know $\mathcal{X}$, and not $\mathcal{W}$. We perform the following procedure on each set $X \in \mathcal{X}$ that contains more than one vertex. Let $\hat{X} = X \cap N_G(V \setminus X)$ be the set of vertices in X that have at least one neighbor outside X. A *shatter* of X is a partition of X into sets, such that one of them is a connected vertex cover C of $G[X]$ containing every vertex of $\hat{X}$, and each of the others has size 1. The *size* of a shatter is the size of C. A shatter of X of minimum size can be found as follows. Recall that we assumed the largest witness set of $\mathcal{W}$ to be of size d. Construct a graph G' from the graph $G[X]$ by adding, for each vertex $x \in \hat{X}$, a new vertex x' and an edge xx'. Find a connected vertex cover C of minimum size in G' by applying the algorithm of Proposition 1 for all values of t from 1 to d. Since ϕ is $\mathcal{W}$-compatible and each witness set of $\mathcal{W}$ has size at most d, such a set C will always be found. Observe that a minimum size connected vertex cover of G' does not contain any vertex of degree 1, which implies that $\hat{X} \subseteq C$. Hence C, together with the sets of size 1 formed by each of the vertices of $X \setminus C$, is a minimum size shatter of X. If, in $\mathcal{X}$, we replace X by the sets of this minimum size shatter of X, we obtain a $\tilde{T}$-witness structure of G, for some tree $\tilde{T}$ with at least as many (or strictly more, if $|C| < |X|$) vertices as T''. We point out that the size of C is at most as big as the size of the only possible big witness set of $\mathcal{W}$ that X contains. Hence, after repeating the above procedure on each of the sets of $\mathcal{X}$ that contain more than one vertex, we obtain a desired T'-witness structure of G, where T' is a tree with at least at many vertices as T.

By Proposition 1, we can find a minimum size shatter in $2.4882^d\, n^{O(1)}$ time for each set of $\mathcal{X}$. Since all the other steps can be performed in polynomial time, the overall running time is $2.4882^d\, n^{O(1)}$. □

The idea of our algorithm for Tree Contraction is to generate a number of 2-colorings of the input graph G, and to check, using the algorithm described in the proof of Lemma 2, whether any of the generated 2-colorings yields a T-witness structure of G for a tree T on at least $n - k$ vertices. Using the notion of universal sets, defined below, we are able to bound the number of 2-colorings that we need to generate and check.

The *restriction* of a function $f : X \to Y$ to a set $S \subseteq X$ is the function $f_{|S} : S \to Y$ such that $f_{|S}(s) = f(s)$ for every $s \in S$. An (n, t)-*universal set* $\mathcal{F}$ is a set of functions from $\{1, 2, \ldots, n\}$ to $\{1, 2\}$ such that, for every $S \subseteq \{1, 2, \ldots, n\}$ with $|S| = t$, the set $\mathcal{F}_{|S} = \{f_{|S} \mid f \in \mathcal{F}\}$ is equal to the set 2^S of all the functions from S to $\{1, 2\}$.

Theorem 2 ([24]). *There is a deterministic algorithm that constructs an* (n, t)*-universal set* $\mathcal{F}$ *of size* $2^{t+O(\log^2 t)} \log n$ *in time* $2^{t+O(\log^2 t)} n \log n$.

Theorem 3. Tree Contraction *can be solved in time* $4.98^k\, n^{O(1)}$.

Proof. Let G be an n-vertex input graph of Tree Contraction. We assume that G is 2-connected, by Lemma 1. Our algorithm has an outer loop, which iterates over the values of an integer d from 1 to $k + 1$. For each value of d, the algorithm constructs an $(n,\ 2k - d + 2)$-universal set $\mathcal{F}_d$, and runs an inner loop that iterates over all 2-colorings $\phi \in \mathcal{F}_d$. At each iteration of the inner loop, the algorithm computes a minimum size shatter for each of the monochromatic components of G with respect to ϕ, using the $2.4882^d\, n^{O(1)}$ time procedure described in the proof of Lemma 2 with the value d determined by the outer loop. If this procedure yields a T'-witness structure of G for a tree T' with at least $n - k$ vertices at some iteration of the inner loop, then the algorithm outputs yes. If none of the iterations of the inner loop yields a yes-answer, the outer loop picks the next value of d. If none of the iterations of the outer loop yields a yes-answer, then the algorithm returns no.

To prove correctness of the algorithm, suppose G is k-contractible to a tree T. Let $\mathcal{W}$ be a T-witness structure of G whose largest witness set has size d^*. Note that $d^* \leq k + 1$ by Observation 1. Let ψ be a 2-coloring of G such that each of the big witness sets of $\mathcal{W}$ is monochromatic with respect to ψ, such that $\psi(W(u)) \neq \psi(W(v))$ whenever uv is an edge in T, and such that the vertices in the small witness sets are all colored 1. Observe that ψ is a $\mathcal{W}$-compatible 2-coloring of G, as is any other 2-coloring of G that coincides with ψ on all the vertices of the big witness sets of $\mathcal{W}$. The largest witness set requires $d^* - 1$ edge contractions, after which our remaining budget of edge contractions is $k - (d^* - 1) = k - d^* + 1$. As a result of Observation 1, the total number of vertices contained in big witness sets is thus at most $d^* + 2(k - d^* + 1) = 2k - d^* + 2$. Consequently, if we generate an $(n,\ 2k - d^* + 2)$-universal set $\mathcal{F}_{d^*}$, then, by Theorem 2, $\mathcal{F}_{d^*}$ contains all the 2-colorings of G that coincide with ψ on all the vertices of the big witness sets of $\mathcal{W}$. In particular, $\mathcal{F}_{d^*}$ contains at least one 2-coloring ϕ that is $\mathcal{W}$-compatible. Recall that our algorithm iterates over all values of d from 1 to $k + 1$, and that $d^* \leq k + 1$. Hence, at the correct iteration of the outer loop, i.e., the iteration where $d = d^*$, our algorithm will process ϕ. As a result of Lemma 2, the algorithm will then find a T'-witness structure of G for some tree T' with at least $n - k$ vertices. This means that the algorithm correctly outputs yes if G is k-contractible to a tree. Since the algorithm only outputs YES when it has detected a T'-witness structure for some tree T' with at least $n-k$ vertices, it correctly outputs NO if G is not k-contractible to a tree.

For each d, the size of $\mathcal{F}_d$ is $2^{2k-d+2+\log^2(2k-d+2)} \log n$, and $\mathcal{F}_d$ can be constructed in $2^{2k-d+2+\log^2(2k-d+2)} n \log n$ time, by Theorem 2. Summing $|\mathcal{F}_d| \cdot 2.4882^d \, n^{O(1)}$ over all values of d from 1 to $k+1$ shows that this deterministic algorithm runs in time $4.98^k \, n^{O(1)}$. $\square$

We would like to remark that due to recent developments in the field, our result in fact implies a randomized $4^{k+o(k)} \, n^{O(1)}$ time algorithm for TREE CONTRACTION. Cygan et al. [14] give a Monte Carlo algorithm with running time $2^t \, n^{O(1)}$ for deciding whether a graph on n vertices has a connected vertex cover of size at most t and finding such a set if it exists. Summing $|\mathcal{F}_d| \cdot 2^d \, n^{O(1)}$ over all values of d from 1 to $k+1$, as it was done in the last line of the proof of Theorem 3, we obtain total running time $4^{k+o(k)} \, n^{O(1)}$ for a randomized algorithm.

4 PATH CONTRACTION

Brouwer and Veldman [9] showed that, for every fixed $\ell \geq 4$, it is NP-complete to decide whether a graph can be contracted to the path P_ℓ. This, together with the observation that a graph G is k-contractible to a path if and only if G is contractible to P_{n-k}, implies that PATH CONTRACTION is NP-complete. In this section, we first show that PATH CONTRACTION has a linear vertex kernel. We then present an algorithm with running time $2^{k+o(k)} + n^{O(1)}$ for this problem. Throughout this section, whenever we mention a P_ℓ-witness structure $\mathcal{W} = \{W_1, \ldots W_\ell\}$, it will be implicit that $P_\ell = p_1 \cdots p_\ell$, and $W_i = W(p_i)$ for every $i \in \{1, \ldots, \ell\}$.

Rule 1 *Let (G, k) be an instance of* PATH CONTRACTION. *If G contains a bridge uv such that the deletion of edge uv from G results in two connected components that contain at least $k+2$ vertices each, then return (G', k), where G' is the graph resulting from the contraction of edge uv.*

Lemma 3. *Let (G', k) be an instance of* PATH CONTRACTION *resulting from the application of Rule 1 on (G, k). Then G' is k-contractible to a path if and only if G is k-contractible to a path.*

Proof. Let G be a graph on which Rule 1 is applicable, and let uv be the bridge of G that is contracted to obtain G'. Let G_1 and G_2 be the two connected components that we obtain if we delete the edge uv from G, with $L = V(G_1)$ and $R = V(G_2)$, such that $u \in L$ and $v \in R$. Furthermore, let $L' = L \setminus \{u\}$ and $R' = R \setminus \{v\}$, and let w be the vertex of G' resulting from the contraction of uv in G.

Assume that G is k-contractible to a path P_ℓ, and let $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ be a P_ℓ-witness structure of G. If u and v belong to the same witness set W_i of $\mathcal{W}$, then we can obtain a P_ℓ-witness structure $\mathcal{W}'$ of G' by replacing W_i with a new set $W_i' = (W_i \setminus \{u, v\}) \cup \{w\}$, and keeping all other witness sets of $\mathcal{W}$ the same. Hence G' can be contracted to P_ℓ. This implies that G' is $(k-1)$-contractible to a path. If u and v belong to two different witness sets of $\mathcal{W}$, then one belongs to W_i and the other to W_{i+1}, for two adjacent vertices p_i and p_{i+1} of P_ℓ. Furthermore, uv is the only edge in G between a vertex of $A = \bigcup_{j=1}^{i} W_j$ and a vertex of $B = \bigcup_{j=i+1}^{\ell} W_j$, since both $G[A]$ and $G[B]$ are connected and uv is a bridge of G. Consequently, by replacing W_i and W_{i+1} by one witness set $W_i'' = ((W_i \cup W_{i+1}) \setminus \{u, v\}) \cup \{w\}$, and keeping all other witness sets of $\mathcal{W}$ the same, we obtain a $P_{\ell-1}$-witness structure $\mathcal{W}''$ of G'. Hence G' is k-contractible to a path.

For the other direction, assume that G' is k-contractible to a path $P_{\ell'}$, and let $\mathcal{W} = \{W_1, \ldots, W_{\ell'}\}$ be a $P_{\ell'}$-witness structure of G'. Let W_i be the witness set of $\mathcal{W}$ containing w. Note that w is a cut vertex of G'. The set $A = \bigcup_{j=1}^{i-1} W_j$ is connected in $G' - w = G - \{u, v\}$. Hence A cannot contain vertices from both L' and R', so it contains only elements of L' or only elements of R'. Similarly, the set $B = \bigcup_{j=i+1}^{\ell} W_j$ contains only elements of L', or only elements of R'. By Observation 1, the set W_i has size at most $k+1$. Hence W_i, which contains w, does not contain all of L', as $|L' \cup \{w\}| \geq k+2$. Similarly, W_i does not contain all of R', as $|R' \cup \{w\}| \geq k+2$. Hence, neither A nor B is empty, one contains only elements of R' and the other only elements of L'. Consequently, by replacing W_i by two new sets $(W_i \cap L') \cup \{u\}$ and $(W_i \cap R') \cup \{v\}$, and keeping all other witness sets of $\mathcal{W}$ the same, we obtain a $P_{\ell'+1}$-witness structure of G. Hence G is k-contractible to a path. □

Theorem 4. Path Contraction *has a kernel with at most* $5k + 3$ *vertices.*

Proof. Let (G, k) be an instance of Path Contraction. We repeatedly test, in linear time, whether Rule 1 can be applied on the instance under consideration, and apply the reduction rule if possible. Each application of Rule 1 strictly decreases the number of vertices. Hence, starting from G, we reach in polynomial time a *reduced* graph, on which Rule 1 cannot be applied anymore. By Lemma 3, we know that the resulting reduced graph is k-contractible to a path if and only if G is k-contractible to a path.

We now assume that G is reduced. We show that if G is k-contractible to a path, then G has at most $5k + 3$ vertices. Let $\mathcal{W} = \{W_1, \ldots, W_\ell\}$ be a P_ℓ-witness structure of G with $\ell \geq n-k$. We first prove that $\ell \leq 4k+3$. Assume that $\ell \geq 2k+4$, and let i be such that $k+2 \leq i \leq \ell-k-2$. Suppose, for contradiction, that both W_i and W_{i+1} are small witness sets, i.e., $W_i = \{u\}$ and $W_{i+1} = \{v\}$ for two vertices u and v of G. Then uv forms a bridge in G whose deletion results in two connected components. Each of these components contains at least all vertices from $W_1, \ldots, W_{k+2}$ or all vertices from $W_{\ell-k-1}, \ldots, W_\ell$. Hence they contain at least $k + 2$ vertices each. Consequently, Rule 1 can be applied, contradicting the assumption that G is reduced. So there are no consecutive small sets among $W_{k+2}, \ldots, W_{\ell-k-1}$. By Observation 1, $\mathcal{W}$ contains at most k big witness sets, so we have $(\ell-k-1)-(k+2)+1 \leq 2k+1$ implying $\ell \leq 4k+3$. Combining this with the earlier assumption that $\ell \geq n-k$ yields $n \leq 5k+3$. □

The existence of a kernel with at most $5k + 3$ vertices easily implies a $32^{k+o(k)} + n^{O(1)}$ time algorithm for Path Contraction, which tests for each

2-coloring ϕ of the reduced input graph whether the monochromatic components of ϕ form a P_ℓ-witness structure of G for some $\ell \geq n-k$. The natural follow-up question is how much we can improve this running time, which we answer next.

Theorem 5. Path Contraction *can be solved in time* $2^{k+o(k)} + n^{O(1)}$.

Proof. Given an instance (G,k) of Path Contraction, our algorithm first constructs an equivalent instance (G',k) such that G' has at most $5k+3$ vertices. This can be done in $n^{O(1)}$ time by Theorem 4. For the rest of the proof, we assume that the input graph G has $n \leq 5k+3$ vertices. Suppose G is k-contractible to a path P_ℓ, and let $\mathcal{W} = \{W_1,\ldots,W_\ell\}$ be a P_ℓ-witness structure of G. We distinguish two cases, depending on whether or not ℓ is larger than $\sqrt{k}$.

Suppose $\ell \leq \sqrt{k}$. Then $n \leq k+\sqrt{k}$. We define $X^* = W_1 \cup W_3 \cup W_5 \cup \ldots$ and $Y^* = W_2 \cup W_4 \cup \ldots$. Then X^* and Y^* form a 2-partition of $V(G)$, and the connected components of the graphs $G[X^*]$ and $G[Y^*]$ form a P_ℓ-witness structure of G. If we contract every edge of G that has both endpoints in the same connected component of $G[X^*]$ or $G[Y^*]$, we end up with the path P_ℓ. Hence, for every given partition of $V(G)$ into two sets X and Y, we can check in $k^{O(1)}$ time whether the connected components of $G[X]$ and $G[Y]$ constitute a $P_{\ell'}$-witness structure of G for some $\ell' \geq n-k$. Based on this analysis, if G has at most $k+\sqrt{k}$ vertices, the algorithm checks for each 2-partition X,Y of $V(G)$ whether this 2-partition yields a desired witness structure. Note that if G is k-contractible to a path on more than $\sqrt{k}$ vertices, then it is also k-contractible to a path on exactly $\sqrt{k}$ vertices, since G has at most $k+\sqrt{k}$ vertices. Since there are at most $2^{k+\sqrt{k}}$ partitions to consider, the running time of the algorithm in this case is $2^{k+o(k)}\, k^{O(1)} = 2^{k+o(k)}$.

Now suppose $\ell > \sqrt{k}$. For each integer i with $1 \leq i \leq \lfloor\sqrt{k}\rfloor$, we define $W_i^* = W_i \cup W_{i+\lfloor\sqrt{k}\rfloor} \cup W_{i+2\lfloor\sqrt{k}\rfloor} \cup \ldots$. Since $n \leq 5k+3$, there is at least one index j such that $|W_j^*| \leq (5k+3)/\sqrt{k}$. Let $G_1^*,\ldots,G_p^*$ denote the connected components of $G - W_j^*$, where $p \leq (5k+3)/\sqrt{k}$. Note that each connected component G_i^* has a P^*-witness structure $\mathcal{W}_i^*$ for some path P^* on at most $\sqrt{k}-1$ vertices, such that the union of these witness structures $\mathcal{W}_i^*$, together with the vertex sets of $G_1^*,\ldots,G_p^*$, forms a P_ℓ-witness structure of G. Moreover, each connected component G_i^* has at most $k+\sqrt{k}-1$ vertices by Observation 1. Based on this analysis, if G has more than $k+\sqrt{k}$ vertices, the algorithm searches for the correct set W_i^* by generating all subsets $W \subseteq V(G)$ of size at most $(5k+3)/\sqrt{k}$, and performing the following checks for each subset W. If the graph $G-W$ has more than $(5k+3)/\sqrt{k}$ connected components, or if one of the connected components has more than $k+\sqrt{k}-1$ vertices, then W is discarded. For each W that is not discarded, we run the algorithm of the previous case on each connected component G_i of $G-W$ to check whether G_i has a $P_{\ell'}$-witness structure with $\ell' \leq \sqrt{k}-1$. Since in that algorithm we check every 2-partition of $V(G_i)$, we can check whether G_i has a $P_{\ell'}$-witness structure with the additional constraint that precisely the vertices in the first and the last witness sets have neighbors in the appropriate connected components of $G[W]$, and pick such a $P_{\ell'}$-witness structure $\mathcal{W}_i$ for which ℓ' is a large as possible. Finally, we check if all these witness structures $\mathcal{W}_i$, together with the vertex sets of the connected components of $G[W]$, form a P-witness structure of G for some path P on at least $n-k$ vertices. If so, the algorithm outputs YES. Otherwise, the algorithm tries another subset W, or outputs NO if all subsets W have been considered. For each generated set W, we run the algorithm of the previous case on each of the $O(\sqrt{k})$ connected components of $G-W$, so we can check in $2^{k+o(k)}O(\sqrt{k}) = 2^{k+o(k)}$ time whether we get a desired P-witness structure of G. Since we generate no more than $(5k+3)^{(5k+3)/\sqrt{k}} = 2^{o(k)}$ subsets W, we get a total running time of $2^{k+o(k)}$ also for this case. □

5 Concluding remarks

The number of edges to contract in order to obtain a certain graph property is a natural measure of how close the input graph is to having that property, similar to the more established similarity measures of the number of edges or vertices to delete. The latter measures are well studied when the desired property is being acyclic or being a path, defining some of the most widely known and well studied problems within Parameterized Complexity. Inspired by this, we gave kernelization results and fast fixed parameter tractable algorithms for PATH CONTRACTION and TREE CONTRACTION. We think our results motivate the parameterized study of similar problems, an example of which is INTERVAL CONTRACTION. It is not known whether the vertex deletion variant of this problem, INTERVAL VERTEX DELETION, is fixed parameter tractable. Is INTERVAL CONTRACTION fixed parameter tractable?

Acknowledgments The authors would like to thank Martin Vatshelle, Saket Saurabh, and Erik Jan van Leeuwen for valuable suggestions and comments.

References

1. N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM* 42:844–856, 1995.
2. T. Asano and T. Hirata. Edge-contraction problems. *J. Comput. Syst. Sci.* 26:197–208, 1983.
3. R. van Bevern, C. Komusiewicz, H. Moser, and R. Niedermeier. Measuring Indifference: Unit Interval Vertex Deletion. In *Proc. WG 2010*, LNCS 6410:232–243, Springer.
4. D. Binkele-Raible and H. Fernau. Enumerate & measure: improving parameter budget management. In *Proc IPEC 2010*, LNCS 6478:38–49, Springer.
5. H. L. Bodlaender. On disjoint cycles. *Int. J. Found. Comput. Sci.* 5:59–68, 1994.
6. H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. (Meta) kernelization. In *Proc. FOCS 2009*, 629-638, IEEE.
7. H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels In *Proc. ICALP 2008*, LNCS 5125:563–574, Springer.
8. H. L. Bodlaender, S. Thomassé, and A. Yeo. Kernel Bounds for Disjoint Cycles and Disjoint Paths. In *Proc. ESA 2009*, LNCS 5757:635-646, Springer.
9. A. E. Brouwer and H. J. Veldman. Contractibility and NP-completeness. *J. Graph Theory* 11(1):71–79, 1987.

10. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.* 58(4):171–176, 1996.
11. Y. Cao, J. Chen, and Y. Liu. On Feedback Vertex Set New Measure and New Structures. In *Proc. SWAT 2010*, LNCS 6139:93-104, Springer.
12. J. Chen, F. V. Fomin, Y. Liu, S. Lu, Y. Villanger. Improved algorithms for feedback vertex set problems, *J. Comput. System Sci.* 74(7):1188–1198, 2008.
13. B. Courcelle. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation* 85:12–75, 1990.
14. M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. Cornell University Library, arXiv:1103.0534, 2011.
15. F. K. H. A. Dehne, M. R. Fellows, M. A. Langston, F. A. Rosamond, K. Stevens. An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem, *Theory Comput. Syst.* 41(3):479–492, 2007.
16. M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through Colors and IDs. In *Proc. ICALP 2009*, LNCS 5555:378–389, Springer.
17. R. G. Downey and M. R. Fellows. *Parameterized complexity.* Monographs in Computer Science, Springer-Verlag, 1999.
18. J. Guo, J. Gramm, F. Huffner, R. Niedermeier, S. Wernicke. Compression-based fixed-parameter algorithms for feedback vertex set and edge bipartization, *J. Comput. Syst. Sci.* 72(8):1386–1396, 2006.
19. P. Heggernes, P. van 't Hof, B. Jansen, S. Kratsch, and Y. Villanger. Parameterized Complexity of Vertex Deletion into Perfect Graph Classes. In *Proc. FCT 2011.* LNCS, Springer. To appear.
20. P. Heggernes, P. van 't Hof, D. Lokshtanov, and C. Paul. Obtaining a bipartite graph by contracting few edges. arXiv:1102.5441, February 2011.
21. K. Kawarabayashi and B. A. Reed. An (almost) linear time algorithm for odd cycles transversal. In Proc. SODA 2010, 365378, ACM-SIAM.
22. D. Marx. Chordal deletion is fixed-parameter tractable. In *Proc. WG 2006*, LNCS 4271:37–48, Springer.
23. D. Marx and I. Schlotter. Obtaining a planar graph by vertex deletion. In *Proc. WG 2007*, LNCS 4769:292–303, Springer.
24. M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *Proc. FOCS 1995*, 182–191, IEEE.
25. A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Disc. Appl. Math* 113(1):109–128, 2001.
26. G. Philip, V. Raman and Y. Villanger. A Quartic Kernel for Pathwidth-One Vertex Deletion. In *Proc. WG 2010*, LNCS 6410:196–207, Springer.
27. S. Thomassé. A quadratic kernel for feedback vertex set. In *Proc. SODA 2009*, 115–119, ACM-SIAM.
28. Y. Villanger. Proper interval vertex deletion. In *Proc. of IPEC 2010*, LNCS 6478:228–238, Springer.
29. T. Watanabe, T. Ae, and A. Nakamura. On the removal of forbidden graphs by edge-deletion or edge-contraction. *Disc. Appl. Math.* 3:151–153, 1981.
30. T. Watanabe, T. Ae, and A. Nakamura. On the NP-hardness of edge-deletion and edge-contraction problems. *Disc. Appl. Math.* 6:63–78, 1983.
31. M. Yannakakis. Node and edge-deletion NP-complete problems. In *Proc. STOC 1978*, 253–264, ACM.
32. M. Yannakakis. The effect of a connectivity requirement on the complexity of maximum subgraph problems. *J. ACM* 26(4):618–630, 1979.
33. M. Yannakakis. Edge-deletion problems. *SIAM J. Comput.* 10(2):297–309, 1981.

Appendix

This appendix contains the proofs that were omitted from the paper due to page restrictions.

Continuation of the proof of Theorem 1. Observe that $k \leq 2|A|$, which means that the construction is parameter preserving. In particular, we added $|A|(k+1)+1 \leq 2|A|^2+|A|+1$ vertices to G to obtain G', and we added $|B|$ edges incident to a and then two edges incident to each vertex of $B' \setminus B$. Hence the size of the graph has increased by $O(|B|+|A|^2)$. We show that there is a subset of at most t vertices in B that dominates A in G if and only if G' is k-contractible to a tree.

Assume there exists a set $S \subseteq B$ of size at most t such that S dominates A in G. Vertex a is adjacent to all vertices of S, so the set $X = \{a\} \cup S \cup A$ is connected in G'. Note that all the vertices of G' that do not belong to X form an independent set in G. Consider the unique witness structure of G' that has X as its only big witness set. Contracting all the edges of a spanning tree of $G[X]$ yields a star. Since X has at most $1+t+|A| = 1+k$ vertices, any spanning tree of $G[X]$ has at most k edges. Hence G' is k-contractible to a tree.

For the reverse direction, assume that G' is k-contractible to a tree T, and let $\mathcal{W}$ be a T-witness structure of G'. Vertex a is involved in $k+1$ different cycles with each vertex of A through the vertices of $B' \setminus B$. Hence, if a and a vertex u of A appear in different witness sets, we need more than k contractions to kill the $k+1$ cycles containing both a and u. Consequently, there must be a witness set $W \in \mathcal{W}$ that contains all the vertices of $A \cup \{a\}$. Since all the vertices of $G' - W$ belong to B', they form an independent set in G'. This means that W is the only big witness set of $\mathcal{W}$, and T is in fact a star. Since G' is k-contractible to T, we know that $|W| \leq k+1$ by Observation 1. Suppose W contains a vertex $x \in B' \setminus B$. By construction, x is adjacent only to a and exactly one vertex $a' \in A$. Let b' be a neighbor of a' in B. Then we have $N_{G'}(x) \subseteq N_{G'}(b')$, so $W' = (W \setminus \{x\}) \cup \{b'\}$ is connected and $|W'| \leq |W|$. The unique witness structure of G' that has W' as its only big witness set shows that G' can be k-contracted to a tree T' on at least as many vertices as T. Thus we may assume that W contains no vertices of $B' \setminus B$. Let $S = W \setminus A'$. The set W is connected and A' is an independent set, so S dominates A'. Moreover $|S| = |W| - |A| - 1 \leq k - |A| = t$. We conclude that S is a subset of at most t vertices in B that dominates A in G. □

Proof of Lemma 1. Let G be a graph, and suppose that G has a vertex v whose deletion results in at least two connected components with vertex sets $V_1, \ldots, V_p$. For every $i \in \{1, \ldots, p\}$, let $G_i = G[V_i \cup \{v\}]$. We prove that G is k-contractible to a tree if and only if each of the graphs G_i is contractible to a tree, using at most k edge contractions in total. Repeating this argument for each of the cut vertices of G yields the lemma.

Suppose each of the graphs G_i can be contracted to a tree T_i, and that at most k edge contractions are used in total; let $E' \subseteq E(G)$ be the corresponding set of contracted edges, with $|E'| \leq k$. For every $i \in \{1, \ldots, p\}$, let $\mathcal{W}_i$ be a T_i-witness structure of the graph G_i, and let W_i be the witness set of $\mathcal{W}_i$ containing v. Note that $W = \bigcup_{i=1}^{p} W_i$ is a connected set in G. We define a witness structure $\mathcal{W}$ of G as follows: $\mathcal{W}$ contains all the witness sets of $\mathcal{W}_i \setminus W_i$ for every $i \in \{1, \ldots, p\}$, as well as one witness set formed by the set W. It is clear that $\mathcal{W}$ is a T-witness structure of G for some tree T, and that contracting each of the edges of E' in G yields T. Since $|E'| \leq k$, G is k-contractible to a tree.

To prove the reverse statement, suppose G is k-contractible to a tree T. Let $\mathcal{W}$ be a T-witness structure of G, and let W be the witness set of $\mathcal{W}$ containing v. Since v is a cut vertex of G, every witness set in $\mathcal{W} \setminus \{W\}$ is contained in exactly one of the sets V_i. For every $i \in \{1, \ldots, p\}$, we define a witness structure $\mathcal{W}_i$ of the graph G_i as follows: $\mathcal{W}_i$ contains every witness set of $\mathcal{W}$ that is contained in V_i, plus one witness set $W_i = (W \cap V_i) \cup \{v\}$. Since v is a cut vertex of G and the set W is connected, each of the sets $W \cap V_i$ is connected as well. We conclude that each graph G_i can be contracted to a tree T_i by repeatedly contracting every edge that has both endpoints in the same witness set of $\mathcal{W}_i$. Since contracting exactly the same edges in G yields the tree T and G is k-contractible to T, the total number of edge contractions needed to contract each G_i to T_i is at most k. $\square$