

# On the Parameterized Complexity of Contraction to Generalization of Trees\*

Akanksha Agrawal<sup>1</sup>, Saket Saurabh<sup>2</sup>, and Prafullkumar Tale<sup>3</sup>

1 Department of Informatics, University of Bergen, Bergen, Norway  
akanksha.agrawal@uib.no

2 Department of Informatics, University of Bergen, Bergen, Norway and  
The Institute of Mathematical Sciences, HBNI, Chennai, India  
saketa@imsc.res.in

3 The Institute of Mathematical Sciences, HBNI, Chennai, India  
pptale@imsc.res.in

---

## Abstract

For a family of graphs  $\mathcal{F}$ , the  $\mathcal{F}$ -CONTRACTION problem takes as an input a graph  $G$  and an integer  $k$ , and the goal is to decide if there exists  $S \subseteq E(G)$  of size at most  $k$  such that  $G/S$  belongs to  $\mathcal{F}$ . Here,  $G/S$  is the graph obtained from  $G$  by contracting all the edges in  $S$ . Heggenes et al. [*Algorithmica* (2014)] were the first to study edge contraction problems in the realm of Parameterized Complexity. They studied  $\mathcal{F}$ -CONTRACTION when  $\mathcal{F}$  is a simple family of graphs such as trees and paths. In this paper, we study the  $\mathcal{F}$ -CONTRACTION problem, where  $\mathcal{F}$  generalizes the family of trees. In particular, we define this generalization in a “parameterized way”. Let  $\mathbb{T}_\ell$  be the family of graphs such that each graph in  $\mathbb{T}_\ell$  can be made into a tree by deleting at most  $\ell$  edges. Thus, the problem we study is  $\mathbb{T}_\ell$ -CONTRACTION. We design an FPT algorithm for  $\mathbb{T}_\ell$ -CONTRACTION running in time  $\mathcal{O}((2\sqrt{\ell} + 2)^{\mathcal{O}(k+\ell)} \cdot n^{\mathcal{O}(1)})$ . Furthermore, we show that the problem does not admit a polynomial kernel when parameterized by  $k$ . Inspired by the negative result for the kernelization, we design a lossy kernel for  $\mathbb{T}_\ell$ -CONTRACTION of size  $\mathcal{O}([k(k+2\ell)]^{\lceil \frac{\alpha}{\alpha-1} \rceil + 1})$ .

**1998 ACM Subject Classification** G.2.2 Graph Algorithms, I.1.2 Analysis of Algorithms

**Keywords and phrases** Graph Contraction, Fixed Parameter Tractability, Graph Algorithms, Generalization of Trees

**Digital Object Identifier** 10.4230/LIPIcs.IPEC.2017.1

## 1 Introduction

Graph editing problems are one of the central problems in graph theory that have been extensively studied in the realm of Parameterized Complexity. Some of the important graph editing operations are vertex deletion, edge deletion, edge addition, and edge contraction. For a family of graphs  $\mathcal{F}$ , the  $\mathcal{F}$ -EDITING problem takes as an input a graph  $G$  and an integer  $k$ , and the goal is to decide whether or not we can obtain a graph in  $\mathcal{F}$  by applying at most  $k$  edit operations on  $G$ . In fact, the  $\mathcal{F}$ -EDITING problem, where the edit operations are restricted to one of vertex deletion, edge deletion, edge addition, or edge contraction have also received a lot of attention in Parameterized Complexity. When we restrict the operations to only deletion operation (vertex/edge deletion) then the corresponding problem is called  $\mathcal{F}$ -VERTEX

---

\* A full paper containing all the proofs and explanations can be found at <https://arxiv.org/abs/1708.00622>



(EDGE) DELETION problem. On the other hand if we only allow edge contraction then the corresponding problem is called  $\mathcal{F}$ -CONTRACTION. The  $\mathcal{F}$ -EDITING problem generalizes several NP-hard problems such as VERTEX COVER, FEEDBACK VERTEX SET, PLANAR  $\mathcal{F}$ -DELETION, INTERVAL VERTEX DELETION, CHORDAL VERTEX DELETION, ODD CYCLE TRANSVERSAL, EDGE BIPARTIZATION, TREE CONTRACTION, PATH CONTRACTION, SPLIT CONTRACTION, CLIQUE CONTRACTION, etc. Most of the studies in the Parameterized Complexity or the classical Complexity Theory have been restricted to combination of vertex deletion, edge deletion or edge addition. Only recently, edge contraction as an edit operation has started to gain attention in the realm of Parameterized Complexity. In this paper, we add another family of graphs  $\mathcal{F}$  – a parameterized generalization of trees – such that  $\mathcal{F}$ -CONTRACTION is fixed parameter tractable (FPT). We also explore the problem from the viewpoints of Kernelization Complexity as well as its new avatar the Lossy Kernelization. For more details on Parameterized Complexity we refer to the books of Downey and Fellows [11, 12], Flum and Grohe [13], Niedermeier [21], and Cygan et al. [8].

Our starting point is the result of Heggenes et al. [17] who studied  $\mathcal{F}$ -CONTRACTION when  $\mathcal{F}$  is the family of paths ( $\mathbb{P}$ ) and trees ( $\mathbb{T}$ ). To the best of our knowledge these were the first results concerning Parameterized Complexity of  $\mathcal{F}$ -CONTRACTION problems. They showed that  $\mathbb{P}$ -CONTRACTION and  $\mathbb{T}$ -CONTRACTION are FPT. Furthermore, they showed that  $\mathbb{T}$ -CONTRACTION does not admit a polynomial kernel. On the other hand  $\mathbb{P}$ -CONTRACTION admits a polynomial kernel with at most  $5k + 3$  vertices (see [18] for an improved bound of  $3k + 4$  on the number of vertices). Moreover,  $\mathcal{F}$ -CONTRACTION is not FPT (unless some unlikely collapse in Parameterized Complexity happens) even for simple family of graphs such as  $P_t$ -free graphs for some  $t \geq 5$ , the family of  $C_t$ -free graphs for some  $t \geq 4$  [6, 19], and the family of split graphs [2]. Here,  $P_t$  and  $C_t$  denotes the path and cycle on  $t$  vertices. In light of these mixed answers, two natural questions are:

1. What additional parameter we can associate with  $\mathbb{T}$ -CONTRACTION such that it admits a polynomial kernel?
2. What additional parameter we can associate with  $\mathbb{T}$ -CONTRACTION such that an FPT algorithm with combination of these parameterizations leads to an algorithm that generalizes the FPT algorithm on trees?

In our earlier paper (a superset of authors) we addressed the first question [1]. In particular we studied  $\mathcal{F}$ -CONTRACTION, where  $\mathcal{F}$  is the *family of trees with at most  $\ell$  leaves* (together with some other problems), and designed a polynomial kernel (hence an FPT algorithm) with  $\mathcal{O}(k\ell)$  vertices. This was complimented by a matching kernel lower bound result. In this paper we focus on the second question.

**Our Problem and Results.** To define our problem formally let us define  $\mathbb{T}_\ell$  to be the family of graphs such that each graph in  $\mathbb{T}_\ell$  can be made into a tree by deleting at most  $\ell$  edges. Thus the problem we study will be called  $\mathbb{T}_\ell$ -CONTRACTION.

$\mathbb{T}_\ell$ -CONTRACTION

**Parameter:**  $k$

**Input:** A graph  $G$  and an integer  $k$ .

**Question:** Does there exist  $S \subseteq E(G)$  of size at most  $k$  such that  $G/S \in \mathbb{T}_\ell$ ?

Observe that for  $\ell = 0$ ,  $\mathbb{T}_\ell$ -CONTRACTION is the usual  $\mathbb{T}$ -CONTRACTION. We design an FPT algorithm for  $\mathbb{T}_\ell$ -CONTRACTION running in time  $\mathcal{O}((2\sqrt{\ell} + 2)^{\mathcal{O}(k+\ell)} \cdot n^{\mathcal{O}(1)})$ . Our algorithm follows the general approach of Heggenes et al. [17] for designing the algorithm for  $\mathbb{T}$ -CONTRACTION. Also, we show that the problem does not admit a polynomial kernel, when parameterized by  $k$ , for any (fixed)  $\ell \in \mathbb{N}$ . Inspired by the negative result on kernelization, we design a lossy kernel for  $\mathbb{T}_\ell$ -CONTRACTION.

**Related Works.** For several families of graphs  $\mathcal{F}$ , early papers by Watanabe et al. [22, 23] and Asano and Hirata [3] showed that  $\mathcal{F}$ -CONTRACTION is NP-complete. From the viewpoint of Parameterized Complexity these problems exhibit properties that are quite different from the problems where the edit operations are restricted to deleting or adding vertices or edges. For instance, deleting  $k$  edges from a graph such that the resulting graph is a tree is polynomial time solvable. On the other hand, Asano and Hirata showed that  $\mathbb{T}$ -CONTRACTION is NP-hard [3]. Furthermore, a well-known result by Cai [5] states that when  $\mathcal{F}$  is a hereditary family of graphs with a finite set of forbidden induced subgraphs then the graph modification problem defined by  $\mathcal{F}$  and the edit operations restricted to vertex deletion, edge deletion, or edge addition admits an FPT algorithm. Moreover, this result does not hold when the edit operation is edge contraction. Lokshtanov et al. [19] and Cai and Guo [6] independently showed that if  $\mathcal{F}$  is either the family of  $P_\ell$ -free graphs for some  $\ell \geq 5$  or the family of  $C_\ell$ -free graphs for some  $\ell \geq 4$  then  $\mathcal{F}$ -CONTRACTION is W[2]-hard. Golovach et al. [14] proved that if  $\mathcal{F}$  is the family of planar graphs then  $\mathcal{F}$ -CONTRACTION is FPT. Belmonte et al. [4] proved that the problem is FPT for  $\mathcal{F}$  being the family of degree constrained graphs like bounded degree, (constant) degenerate and (constant) regular graphs. Moreover, Cai and Guo [6] showed that in case  $\mathcal{F}$  is the family of cliques,  $\mathcal{F}$ -CONTRACTION is solvable in time  $2^{\mathcal{O}(k \log k)} \cdot n^{\mathcal{O}(1)}$ , while in case  $\mathcal{F}$  is the family of chordal graphs, the problem is W[2]-hard. Heggernes et al. [16] developed an FPT algorithm for the case where  $\mathcal{F}$  is the family of bipartite graphs (see [15] for a faster algorithm).

## 2 Preliminaries

In this section, we state some basic definitions and introduce terminologies from graph theory and algorithms. We also establish some of the notations that will be used throughout. We denote the set of natural numbers by  $\mathbb{N}$ . For  $k \in \mathbb{N}$ , by  $[k]$  we denote the set  $\{1, 2, \dots, k\}$ .

**Graphs.** We use standard terminologies from the book of Diestel [10] for the graph related terms which are not explicitly defined here. We consider simple graphs. For a graph  $G$ , by  $V(G)$  and  $E(G)$  we denote the vertex and edge sets of  $G$ , respectively. For a vertex  $v \in V(G)$ , we use  $\deg_G(v)$  to denote the degree of  $v$  in  $G$ , *i.e.* the number of edges in  $G$  that are incident to  $v$ . For  $v \in V(G)$ , by  $N_G(v)$  we denote the set  $\{u \in V(G) \mid vu \in E(G)\}$ . We drop the subscript  $G$  from  $\deg_G(v)$  and  $N_G(v)$  whenever the context is clear. For a vertex subset  $S \subseteq V(G)$ , by  $G[S]$  we denote the graph with the vertex set  $S$  and the edge set as  $\{vu \in E(G) \mid v, u \in S\}$ . By  $G - S$  we denote the graph  $G[V(G) \setminus S]$ . We say  $S, S' \subseteq V(G)$  are *adjacent* if there is  $v \in S$  and  $v' \in S'$  such that  $vv' \in E(G)$ . Further, an edge  $uv \in E(G)$  is *between*  $S$  and  $S'$  if  $u \in S$  and  $v \in S'$ . For  $E' \subseteq E(G)$ , by  $G/E'$  we denote the graph obtained from  $G$  by contracting the edges in  $E'$ . For  $\ell \in \mathbb{N}$ , by  $\mathbb{T}_\ell$  we denote the family of graphs from which we can obtain a tree using at most  $\ell$  edge deletions. Observe that for any graph  $G \in \mathbb{T}_\ell$ , we have  $|E(G)| \leq |V(G)| - 1 + \ell$ . Moreover, for any connected graph  $G$ , if  $|E(G)| \leq |V(G)| - 1 + \ell$  then  $G \in \mathbb{T}_\ell$ .

A graph  $G$  is *contractible* to a graph  $H$ , if there exists  $E' \subseteq E(G)$  such that  $G/E'$  is *isomorphic* to  $H$ . In other words,  $G$  is *contractible* to  $H$  if there exists a *surjective* function  $\varphi : V(G) \rightarrow V(H)$  with  $W(h) = \{v \in V(G) \mid \varphi(v) = h\}$ , for  $h \in V(H)$  and the following property holds.

- For all  $h, h' \in V(H)$ ,  $hh' \in E(H)$  if and only if  $W(h), W(h')$  are *adjacent* in  $G$ .
- For all  $h \in V(H)$ ,  $G[W(h)]$  is *connected*.

Let  $\mathcal{W} = \{W(h) \mid h \in V(H)\}$ . Observe that  $\mathcal{W}$  defines a partition of vertices in  $G$ . We call  $\mathcal{W}$  as an  $H$ -witness structure of  $G$ . The sets in  $\mathcal{W}$  are called *witness sets*. If a *witness set* contains more than one vertex then we will call it a *big* witness-set, otherwise it is a *small* witness set. A graph  $G$  is said to be  $k$ -contractible to a graph  $H$  if there exists  $E' \subseteq E(G)$  such that  $G/E'$  is isomorphic to  $H$  and  $|E'| \leq k$ .

For a subset  $S \subseteq V(G)$  and a  $k$ -coloring  $\phi$  of  $G$ ,  $S$  is said to be *monochromatic* with respect to  $\phi$  if for all  $s, s' \in S$ ,  $\phi(s) = \phi(s')$ . Observe that  $\phi$  partitions  $V(G)$  into (at most)  $k$  pairwise disjoint sets. A subset  $S \subseteq V(G)$  is said to be *monochromatic component* with respect to  $\phi$  if  $S$  is monochromatic and  $G[S]$  is connected.

### 3 FPT Algorithm for $\mathbb{T}_\ell$ -Contraction

In this section, we design an FPT algorithm for  $\mathbb{T}_\ell$ -CONTRACTION. Our algorithm proceeds as follows. We start by applying some simple reduction rules. Then by branching we ensure that the resulting graph is 2-connected. Finally, we give an FPT algorithm running in time  $\mathcal{O}((2\sqrt{\ell} + 2)^{\mathcal{O}(k+\ell)} \cdot n^{\mathcal{O}(1)})$  on 2-connected graphs. The approach we use for designing the algorithm for the case when the input graph is 2-connected follows the approach of Heggernes et al. [17] for designing an FPT algorithm for contracting to trees. Also, whenever we are dealing with an instance of  $\mathbb{T}_\ell$ -CONTRACTION we assume that we have an algorithm running in time  $\mathcal{O}((2\sqrt{\ell'} + 2)^{\mathcal{O}(k+\ell')} \cdot n^{\mathcal{O}(1)})$  for  $\mathbb{T}_{\ell'}$ -CONTRACTION, for every  $\ell' < \ell$ . That is, we give family of algorithms inductively for each  $\ell' \in \mathbb{N}$ , where the algorithm for TREE CONTRACTION by Heggernes et al. forms the base case of our inductive hypothesis.

We start with few observation regarding the graph class  $\mathbb{T}_\ell$ , which will be useful while designing the algorithm.

► **Observation 1.** For each  $T \in \mathbb{T}_\ell$  the following statements hold.

1. The chromatic number of  $T$  is at most  $2\sqrt{\ell} + 2$ .
2. If  $T'$  is a graph obtained by subdividing an edge in  $T$  then  $T' \in \mathbb{T}_\ell$ .
3. If  $T'$  is a graph obtained by contracting an edge in  $T$  then  $T' \in \mathbb{T}_\ell$ .

Let  $(G, k)$  be an instance of  $\mathbb{T}_\ell$ -CONTRACTION. The measure we use for analysing the running time of our algorithm is  $\mu = \mu(G, k) = k$ . We start by applying some simple reduction rules.

► **Reduction Rule 3.1.** If  $k < 0$  then return that  $(G, k)$  is a no instance of  $\mathbb{T}_\ell$ -CONTRACTION.

► **Reduction Rule 3.2.** If  $k = 0$  and  $G \in \mathbb{T}_\ell$  then return that  $(G, k)$  is a yes instance of  $\mathbb{T}_\ell$ -CONTRACTION.

► **Reduction Rule 3.3.** If  $G$  is a disconnected or  $k = 0$  and  $G \notin \mathbb{T}_\ell$  then return that  $(G, k)$  is a no instance.

We assume that the input graph is 2-connected, and design an algorithm for input restricted to 2-connected graphs. Later, we will show how we can remove this constraint. The key idea behind the algorithm is to use a coloring of  $V(G)$  with at most  $2\sqrt{\ell} + 2$  colors to find a  $T$ -witness structure (if it exists) of  $G$ , where  $G$  is contractible to  $T \in \mathbb{T}_\ell$  using at most  $k$  edge contractions (see Observation 1). Moreover, if such a  $T$  does not exist then we must correctly conclude that  $(G, k)$  is a no instance of  $\mathbb{T}_\ell$ -CONTRACTION. Towards this, we introduce the following notion.

► **Definition 2.** Let  $G$  be a 2-connected graph,  $T$  be a graph in  $\mathbb{T}_\ell$ ,  $\mathcal{W}$  be a  $T$ -witness structure of  $G$ , and  $\phi : V(G) \rightarrow [2\sqrt{\ell} + 2]$  be a coloring of  $V(G)$ . Furthermore, let  $T_S$  be a

(fixed) spanning tree of  $T$ ,  $M = \{t, t' \mid tt' \in E(T) \setminus E(T_S)\} \cup \{t \in V(T) \mid d_T(t) \geq 3\}$ , and  $B = \{t \in V(T) \mid |W(t)| \geq 2\}$ . We say that  $\phi$  is  $\mathcal{W}$ -compatible if the following conditions are satisfied.

1. For all  $W \in \mathcal{W}$ , and  $w, w' \in W$  we have  $\phi(w) = \phi(w')$ .
2. For all  $t, t' \in M \cup B$  such that  $tt' \in E(T)$  we have  $\phi(W(t)) \neq \phi(W(t'))$ .

We refer to the set  $M \cup B$  as the set of marked vertices.

Assume that  $(G, k)$  is a *yes* instance of  $\mathbb{T}_\ell$ -CONTRACTION, and  $F$  be one of its (inclusion-wise) minimal solution. Furthermore, let  $T = G/F$ , and  $\mathcal{W}$  be the  $T$ -witness structure of  $G$ . Suppose we are given  $G$  and a  $\mathcal{W}$  compatible coloring  $\phi : V(G) \rightarrow [2\sqrt{\ell} + 2]$  of  $G$ , but we are neither given  $\mathcal{W}$  nor  $T$ . We will show how we can compute a  $T'$  witness structure  $\mathcal{W}'$  of  $G$  such that  $|V(T')| \geq |V(T)|$ , where  $T' \in \mathbb{T}_\ell$ . Informally, we will find such a witness structure by either concluding that none of the edges are part of the solution, some specific set of edges are part of the solution, or finding a star-like structure of the monochromatic components of size at least 2 in  $G$ , with respect to  $\phi$ . Towards this, we will employ the algorithm for CONNECTED VERTEX COVER (CVC) by Cygan [7].

► **Proposition 3** ([7]). *CVC admits an algorithm running in time  $2^k n^{\mathcal{O}(1)}$ . Here,  $k$  is the size of a solution and  $n$  is the number of vertices in the input graph.*

We note that we use the algorithm of Cygan [7] instead of the algorithm by Cygan et al. [9], because the latter algorithm is a randomized algorithm. Also, the algorithm given by Proposition 3 can be used to output a solution.

Consider the case when  $G$  is  $k$ -contractable to a graph, say  $T \in \mathbb{T}_\ell$ , and let  $\mathcal{W}$  be a  $T$ -witness structure of  $G$ . Furthermore, let  $\phi : V(G) \rightarrow [2\sqrt{\ell} + 2]$  be a  $\mathcal{W}$ -compatible coloring of  $G$ , and  $\mathcal{X}$  be the set of monochromatic components of  $\phi$ . We prove some lemmata showing useful properties of  $\mathcal{X}$ .

► **Lemma 4.** *Let  $T'$  be the graph with  $\mathcal{X}$  as the  $T'$ -witness structure of  $G$ . Then  $T' \in \mathbb{T}_\ell$  and  $|V(T')| \leq |V(T)|$ .*

Next, we proceed to show how we can partition each  $X \in \mathcal{X}$  into many smaller witness sets such that either we obtain  $\mathcal{W}$  or a  $T'$ -witness structure of  $G$  for some  $T' \in \mathbb{T}_\ell$  which has at least as many vertices as  $T$ . Towards this, we introduce the following notions.

For  $X \in \mathcal{X}$ , by  $\hat{X}$  we denote the set of vertices that have a neighbor outside of  $X$ , *i.e.*  $\hat{X} = N(V(G) \setminus X)$ . 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 all the vertices in  $\hat{X}$  and all other sets are of size 1. The size of a shatter of  $X$  is the size of  $C$ . Furthermore, a shatter of  $X$  is minimum if there is no other shatter with strictly smaller size.

From Lemma 4 (and Definition 2) it follows that for each  $X \in \mathcal{X}$  there is  $\mathcal{W}_X \subseteq \mathcal{W}$  such that  $X = \cup_{Y \in \mathcal{W}_X} Y$ . In the following lemma, we prove some properties of sets in  $\mathcal{W}_X$ , which will be useful in the algorithm design.

► **Lemma 5.** *Consider  $X \in \mathcal{X}$  with  $|X| \geq 2$ ,  $\mathcal{W}_X \subseteq \mathcal{W}$  such that  $X = \cup_{Y \in \mathcal{W}_X} Y$ , and all of the following conditions are satisfied.*

- $G[X] = (u, v_1, \dots, v_q, v)$  is an induced path, where  $q \in \mathbb{N}$ .
- For each  $i \in [q]$  we have  $\deg(v_i) = 2$ .
- There exists  $X' \in \mathcal{X} \setminus \{X\}$  such that  $N(u) \cap X' \neq \emptyset$  and  $N(v) \cap X' \neq \emptyset$ .

Then  $|\mathcal{W}_X| = 1$ .

► **Lemma 6.** *Consider  $X \in \mathcal{X}$  with  $|X| \geq 2$ ,  $\mathcal{W}_X$  such that  $X = \cup_{Y \in \mathcal{W}_X} Y$ , and all the following conditions are satisfied.*

- $G[X] = (v_0, v_1, \dots, v_q, v)$  is an induced path, where  $q \in \mathbb{N}$ .
  - For each  $i \in [q]$  we have  $\deg(v_i) = 2$ .
  - There exists no  $X' \in \mathcal{X}$  such that  $N(u) \cap X' \neq \emptyset$  and  $N(v) \cap X' \neq \emptyset$ .
- Then  $|\mathcal{W}_X| = |X|$ .

Next, we show that each  $X \in \mathcal{X}$  for which Lemma 5 and 6 are not applicable must contain exactly one big witness set. Moreover, the unique big witness set (together with other vertices as singleton sets) forms one of its shatters.

► **Lemma 7.** *For  $X \in \mathcal{X}$  with  $|X| \geq 2$ , let  $\mathcal{W}_X \subseteq \mathcal{W}$  such that  $X = \cup_{Y \in \mathcal{W}_X} Y$ . Furthermore, the set  $X$  does not satisfy the conditions of Lemma 5 or 6. Then there is exactly one big witness set in  $\mathcal{W}_X$ .*

► **Lemma 8.** *Consider  $X \in \mathcal{X}$  such that  $|X| \geq 2$  and it contains a big witness set, and it does not satisfy conditions of Lemma 5 or 6. Let  $\mathcal{W}_X \subseteq \mathcal{W}$  such that  $X = \cup_{Y \in \mathcal{W}_X} Y$ , and  $W^*$  be the (unique) big witness set in  $X$ . Then  $W^*$  is a connected vertex cover of  $G[X]$  and it contains  $\hat{X}$ .*

Using Lemma 6 to Lemma 8 we show how we can replace each  $X \in \mathcal{X}$  with the sets of its shatter. Recall that we are given only  $G$  and  $\phi$ , and therefore we know  $\mathcal{X}$ , but we do not know  $\mathcal{W}$ . In the Lemma 9, we show how we can find a  $T'$ -witness structure of  $G$  for some  $T' \in \mathbb{T}_\ell$ , which has at least as many vertices as  $T$  (without knowing  $\mathcal{W}$ ).

► **Lemma 9.** *Given  $\mathcal{X}$ , we can obtain a  $T'$ -witness structure of  $G$  in time  $2^k n^{\mathcal{O}(1)}$  time, where  $T' \in \mathbb{T}_\ell$  and  $|V(T')| \geq |V(T)|$ .*

Now we are ready to present our randomized algorithm for  $\mathbb{T}_\ell$ -CONTRACTION when input graph is 2-connected.

► **Theorem 10.** *There is a Monte Carlo algorithm for solving  $\mathbb{T}_\ell$ -CONTRACTION on 2-connected graphs running in time  $\mathcal{O}((2\sqrt{\ell} + 2)^{\mathcal{O}(k+\ell)} \cdot n^{\mathcal{O}(1)})$ , where  $n$  is the number of vertices in the input graph. It does not return false positive and returns correct answer with probability at least  $1 - 1/e$ .*

**Proof.** Let  $(G, k)$  be an instance of  $\mathbb{T}_\ell$ -CONTRACTION, where  $G$  is a 2-connected graph. Furthermore, the Reduction Rules 3.1 and 3.3 are not applicable, otherwise we can correctly decide whether or not  $(G, k)$  is a *yes* instance. The algorithm starts by computing a random coloring  $\phi : V(G) \rightarrow [2\sqrt{\ell} + 2]$ , by choosing a color for each vertex uniformly and independently at random. Let  $\mathcal{X}$  be the set of monochromatic connected components with respect to  $\phi$  in  $G$ . The algorithm applies Lemma 9 in time  $2^k n^{\mathcal{O}(1)}$  and tries to compute  $T'$  such that  $T' \in \mathbb{T}_\ell$  and  $G$  is  $k$ -contractible to  $T'$ . It runs  $(2\sqrt{\ell} + 2)^{6k+8\ell}$  many iterations of two steps mentioned above. If for any such iteration it obtains a desired  $T'$ -witness structure of  $G$  then it returns *yes*. If none of the iterations yield *yes* then the algorithm returns *no*. This completes the description of the algorithm.

Observe that the algorithm returns *yes* only if it has found a  $T' \in \mathbb{T}_\ell$  such that  $G$  is contractible to  $T'$  using at most  $k$  edge contractions. Therefore, when it outputs *yes*, then indeed  $(G, k)$  is a *yes* instance of  $\mathbb{T}_\ell$ -CONTRACTION. We now argue that if  $(G, k)$  is a *yes* instance then using a random coloring the algorithm (correctly) returns the answer with sufficiently high probability. Let  $T$  be a graph in  $\mathbb{T}_\ell$ , such that  $G$  is  $k$ -contractible to  $T$ , and  $\mathcal{W}$  be a  $T$ -witness structure of  $G$ . Furthermore, let  $T_S$  be a (fixed) spanning tree of  $T$ , and vertex set  $M, B$  are set of vertices defined in Definition 2. Let  $\psi : V(G) \rightarrow [2\sqrt{\ell} + 2]$  be a coloring where colors are chosen uniformly at random for each vertex. The total number

of vertices contained in big witness sets of  $\mathcal{W}$  is at most  $2k$ . By our assumption, every leaf is a singleton witness set and it is adjacent to a big witness set. Here, we assume that the number of vertices in  $T$  is at least 3, otherwise we can solve the problem in polynomial time. This implies that no leaf is in  $M \cup B$ . Consider graph  $T'$  obtained from  $T$  by deleting all the leaves and deleting edges in  $E(T_\ell) \setminus E(T_S)$ . All the marked vertices of  $T_\ell$  and all the paths connecting two marked vertices are also present in  $T'$ . Notice that  $T'$  is tree with at most  $k + 2\ell$  leaves. Since the number of vertices of degree three is at most the number of leaves in any tree, there are at most  $k + 2\ell$  vertices of degree at least 3. There are at most  $k$  vertices in  $T$  which are big witness sets and at most  $2\ell$  vertices incident to edges in  $E(T_\ell) \setminus E(T_S)$ . Hence the total number of marked vertices is at most  $2k + 4\ell$ . Since  $T'$  is a tree, there are at most  $2k + 4\ell$  vertices which lie on a path between two vertices in  $M \cup B$  and are adjacent to one of these. The number of vertices of  $G$  which are marked vertices or vertices which are adjacent to it in  $T'$  is at most  $2(2k + 4\ell) + 2k$ . Therefore, the probability that  $\psi$  is compatible with  $\mathcal{W}$  is at least  $1/(2\sqrt{\ell} + 2)^{6k+8\ell}$ . Since the algorithm runs  $(2\sqrt{\ell} + 2)^{6k+8\ell}$  many iterations, probability that none of these colorings which is generated uniformly at random is compatible with  $\mathcal{W}$  is at most  $(1 - 1/(2\sqrt{\ell} + 2)^{6k+8\ell})^{(2\sqrt{\ell} + 2)^{6k+8\ell}} < 1/e$ . Hence, algorithm returns a solution on positive instances with probability at least  $1 - 1/e$ . Each iteration takes  $2^k \cdot n^{\mathcal{O}(1)}$  time and hence the total running time of the algorithm is  $\mathcal{O}((2\sqrt{\ell} + 2)^{\mathcal{O}(k+\ell)} \cdot n^{\mathcal{O}(1)})$ . ◀

Next, we design reduction rules and a branching rule whose (exhaustive) application will ensure that the instance of  $\mathbb{T}_\ell$ -CONTRACTION we are dealing with is 2-connected. Either we apply one of these reduction rules or branching rule, or we resolve the instance using the algorithm for  $\mathbb{T}_{\ell'}$ -CONTRACTION, where  $\ell' < \ell$ . This together with Theorem 10 gives us an algorithm for  $\mathbb{T}_\ell$ -CONTRACTION on general graphs.

► **Lemma 11.** *If for some  $0 \leq \ell' < \ell$ ,  $(G, k)$  is a yes instance of  $\mathbb{T}_{\ell'}$ -CONTRACTION then return that  $(G, k)$  is a yes instance of  $\mathbb{T}_\ell$ -CONTRACTION.*

Our next reduction rule deals with vertices of degree of 1.

► **Reduction Rule 3.4.** *If there is  $v \in V(G)$  such that  $d(v) = 1$  then delete  $v$  from  $G$ . The resulting instance is  $(G - \{v\}, k)$ .*

If a connected graph  $G$  is not 2-connected graph then there is a cut vertex say,  $v$  in  $G$ . Let  $C_1, C_2, \dots, C_t$  be the components of  $G - \{v\}$ . Furthermore, let  $G_1 = G[V(C_1) \cup \{v\}]$  and  $G_2 = G - V(C_1)$ . Next, we try to resolve the instance (if possible) using the following lemma.

► **Lemma 12.** *If there exists  $\ell_1$  and  $\ell_2$  with  $\ell_1 + \ell_2 = \ell$ , where  $\ell_1, \ell_2 > 0$ , and  $k_1$  and  $k_2$  with  $k_1 + k_2 = k$  such that  $(G_1, k_1)$  is a yes instance of  $\mathbb{T}_{\ell_1}$ -CONTRACTION and  $(G_2, k_2)$  is a yes instance of  $\mathbb{T}_{\ell_2}$ -CONTRACTION then return that  $(G, k)$  is a yes instance of  $\mathbb{T}_\ell$ -CONTRACTION.*

Notice that if Lemma 12 is not applicable then one of  $G_1$  or  $G_2$  must be contracted to a tree. Let  $k_1$  be the smallest integer such that  $(G_1, k_1)$  is a yes instance of  $\mathbb{T}$ -CONTRACTION, and  $k_2$  be the smallest integer such that  $(G_2, k_2)$  is a yes instance of  $\mathbb{T}$ -CONTRACTION. Notice that  $k_1$  and  $k_2$  can be computed in (deterministic) time  $4^k n^{\mathcal{O}(1)}$  using the algorithm for  $\mathbb{T}$ -CONTRACTION [17]. We next proceed with the following branching rule.

► **Branching Rule 3.1.** *We branch depending on which of the graphs among  $G_1$  and  $G_2$  are contracted to a tree. Therefore, we branch as follows.*

- Contract  $G_1$  to a tree, and the resulting instance is  $(G_2, k - k_1)$ .
- Contract  $G_2$  to a tree, and the resulting instance is  $(G_1, k - k_2)$ .

Note that the measure strictly decreases in each of the branches of the Branching Rule 3.1 since Reduction Rule 3.4 is not applicable. If we are unable to resolve the instance using Lemma 11 and 12, and Reduction Rules 3.3 and 3.4 and Branching Rule 3.1 are not applicable then the input graph is 2-connected. And, then we resolve the instance using Theorem 10.

► **Theorem 13.** *For each  $\ell \in \mathbb{N}$ , there is a Monte Carlo algorithm for solving  $\mathbb{T}_\ell$ -CONTRACTION with running in time  $\mathcal{O}((2\sqrt{\ell} + 2)^{\mathcal{O}(k+\ell)} \cdot n^{\mathcal{O}(1)})$ . It does not return false positive and returns correct answer with probability at least  $1 - 1/e$ .*

## 4 Derandomization

In this section, we derandomize the algorithm presented in Section 3. Before proceeding forward we define the following important object of this section.

► **Definition 14 (Universal Family).** A  $(n, k, q)$ -universal family is a collection  $\mathcal{F}$ , of functions from  $[n]$  to  $[q]$  such that for each  $S \subseteq [n]$  of size  $k$  and a function  $\phi : S \rightarrow [q]$ , there exists function  $f \in \mathcal{F}$  such that  $f|_S \equiv \phi$ .

Here,  $f|_S$  denotes the function  $f$  when restricted to the elements of  $S$ . For  $q = 2$ , the universal family defined above is called an  $(n, k)$ -universal set [20]. Hence,  $(n, k, q)$ -universal family is a generalization of  $(n, k)$ -universal set. The main result of this section is the following theorem (Theorem 15), which we use to derandomize the algorithm presented in Section 3.

► **Theorem 15.** *For any  $n, k, q \geq 1$ , one can construct an  $(n, k, q)$ -universal family of size  $\mathcal{O}(q^k \cdot k^{\mathcal{O}(k)} \cdot \log n)$  in time  $\mathcal{O}(q^k \cdot k^{\mathcal{O}(k)} \cdot n \log n)$ .*

Before proceeding to the proof of Theorem 15, we state how we use it to derandomize the algorithm presented in Section 3. Let  $(G, k)$  be an instance of  $\mathbb{T}_\ell$ -CONTRACTION. Assume that  $(G, k)$  is a *yes* instance of  $\mathbb{T}_\ell$ -CONTRACTION, and let  $F$  be one of its solution. Furthermore, let  $T = G/F$ , where  $T \in \mathbb{T}_\ell$  and  $\mathcal{W}$  be the  $T$ -witness structure of  $G$ , and  $\phi : V(G) \rightarrow [2\sqrt{\ell} + 2]$  be a  $\mathcal{W}$ -compatible coloring of  $G$ . Recall that our randomized algorithm starts by coloring vertices in  $G$  uniformly and independently at random, and then uses this coloring to extract a witness structure out of each color classes. We then argued that any random coloring is “equally good” as that of  $\phi$  with sufficiently high probability, which is given by a function of  $k$  (and  $\ell$ ). To derandomize this algorithm, we construct a family  $\mathcal{F}$  of (coloring) functions from  $[n]$  to  $[2\sqrt{\ell} + 2]$ . We argue that one of the colorings in the family that we compute is “equally good” as that of  $\phi$ . Recall that the number of vertices which we need to be colored in a specific way for a coloring to be  $\mathcal{W}$ -compatible is bounded by  $6k + 8\ell$  (see Definition 2 and Theorem 10). Let  $S$  be the set of vertices in  $G$  which needs to be colored in a specific way as per the requirements of Definition 2. We can safely assume that  $|S| = 6k + 8\ell$ . If this is not the case we can add arbitrary vertices in  $S$  to ensure this. Notice that any coloring  $f$  of  $G$  such that  $f|_S = \phi|_S$  also satisfies the requirements of Definition 2. Let  $\mathcal{F}$  be an  $(n, 6k + 8\ell, 2\sqrt{\ell} + 2)$ -universal family constructed using Theorem 15. Instead of using random coloring in the algorithm presented in Section 3, we can iterate over functions in  $\mathcal{F}$ . Notice that we do not know  $S$  but for any such  $S$ , we are guaranteed to find an appropriate coloring in one of the functions in  $\mathcal{F}$ , which gives us the desired derandomization of the algorithm.

In rest of the section, we focus on the prove of Theorem 15. Overview of the proof is as follows: Let  $S$  be a set of size  $k$  in an  $n$ -sized universe  $U$ . We first *reduce* this universe  $U$  to another universe  $U'$  whose size is bounded by  $k^2$ . We ensure that all elements of  $S$  are mapped to different elements of  $U'$  during this reduction. Let  $Y$  be the range of  $S$  in  $U'$ .



We further partition  $U'$  into  $\log k$  parts such that  $Y$  is almost equally divided among these partition. In other words, each partition contains (roughly)  $k/\log k$  many elements of  $Y$ . For each of these parts, we explicitly store functions which represents all possible  $q$ -coloring of elements of  $Y$  in this partition. Finally, we “pull back” these functions to obtain a coloring of  $S$ .

► **Definition 16** (Splitter [20]). An  $(n, k, q)$ -splitter  $\mathcal{F}$  is a family of functions from  $[n]$  to  $[q]$  such that for every set  $S \subseteq [n]$  of size  $k$  there exists a function  $f \in \mathcal{F}$  that splits  $S$  evenly. That is, for every  $1 \leq z, z' \leq q$ ,  $|f^{-1}(z) \cap S|$  and  $|f^{-1}(z') \cap S|$  differ by at most 1.

► **Lemma 17.** For every  $1 \leq k, q \leq n$  there is a family of  $(n, k, q)$ -splitter of size  $\mathcal{O}(n^{\mathcal{O}(q)})$  which can be constructed in the same time.

Following is another well known result for construction of splitter when  $q = k^2$ . We use this result to *reduce* the size of the universe.

► **Proposition 18** ([8, 20]). For any  $n, k \geq 1$  one can construct an  $(n, k, k^2)$ -splitter of size  $\mathcal{O}(k^{\mathcal{O}(1)} \log n)$  in time  $\mathcal{O}(k^{\mathcal{O}(1)} n \log n)$ .

Next, we look at the  $k$ -RESTRICTION problem defined by Naor et al. [20]. Before defining the problem, we define some terminologies that will be useful. For a fixed set of alphabets, say  $\{1, 2, \dots, b\}$  and a vector *vector*  $V$ , which is an ordered collection of alphabets, the length of  $V$  is the size of the collection. We represent  $n$  length vector  $V$  as  $(v_1, v_2, \dots, v_n)$ . For a positive integer  $i \in [n]$ ,  $V[i]$  denotes the alphabet at the  $i^{\text{th}}$  position of  $V$ . Similarly, for an (index) set  $S \subseteq [n]$ ,  $V[S]$  denotes the  $|S|$  sized vector obtained by taking alphabet at  $i^{\text{th}}$  position in  $V$ , for each  $i \in S$ . In other words, if  $S = \{i_1, i_2, \dots, i_k\}$  for  $i_1 < i_2 < \dots < i_k$ , then  $V[S] = (V[i_1], V[i_2], \dots, V[i_k])$ . An input to the  $k$ -RESTRICTION problem is a set  $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$  called as a  $k$ -restrictions, where  $C_j \subseteq [b]^k$  for  $j \in [m]$  and an integer  $n$ . Here,  $[b]^k$  denotes the set of all possible vectors of length  $k$  over  $[b]$ , and  $m$  denotes the size of the  $k$ -restrictions. We say that a collection  $\mathcal{V}$  of vectors *obeys*  $\mathcal{C}$  if for all  $S \subseteq [n]$  which is of size  $k$  and for all  $C_j \in \mathcal{C}$ , there exists  $V \in \mathcal{V}$  such that  $V[S] \in C_j$ . The goal of  $k$ -RESTRICTION problem is to find a collection  $\mathcal{V}$  of as small cardinality as possible, which obeys  $\mathcal{C}$ . Let  $c = \min_{j \in [m]} |C_j|$ , and let  $T$  be the time needed to check whether or not the vector  $V$  is in  $C_j$ . We next state the result of Naor et al. [20], which will be useful for proving Theorem 15.

► **Proposition 19** (Theorem 1 [20]). For any  $k$ -RESTRICTION problem with  $b \leq n$ , there is a deterministic algorithm that outputs a collection obeying  $k$ -restrictions, which has size at most  $(k \log n + \log m) / \log(b^k / (b^k - c))$ . Moreover, the algorithm runs in time  $\mathcal{O}(\frac{b^k}{c} \binom{n}{k} \cdot m \cdot T \cdot n^k)$ . Here,  $b$  is the size of the alphabet set,  $m$  is the size of the  $k$ -restrictions,  $n$  is the size of the vectors in the output set, and  $c$  is the size of the smallest collection in the  $k$ -restrictions.

Notice that a function from  $[n]$  to  $[q]$  can be seen as an  $n$ -length vector over the alphabet set  $[q]$ . Consider the case when each  $C_j$  contains exactly one vector of length  $k$  over  $[q]$ , i.e.  $\mathcal{C} = \{\{C\} \mid C \in [q]^k\}$ ,  $m = q^k$ ,  $c = 1$ , and  $T = \mathcal{O}(n)$ . The output of  $k$ -RESTRICTION on this input is exactly an  $(n, k, q)$ -universal family. Therefore, we obtain the following corollary.

► **Corollary 20.** For any  $n, k, q \geq 1$ , one can construct an  $(n, k, q)$ -universal family of size  $\mathcal{O}(q^k \cdot k \cdot (\log n + \log q))$  in time  $\mathcal{O}(q^k \cdot n^{\mathcal{O}(k)})$ .

Notice that we can not directly employ Corollary 20 to construct the desired family, since its running time is  $\mathcal{O}(q^k \cdot n^{\mathcal{O}(k)})$ . Therefore, we carefully use splitter to construct an  $(n, k, q)$ -universal family to obtain the desired running time.

*Proof of Theorem 15.* For the sake of clarity in the notations, we assume that  $\log k$  and  $k/\log k$  are integers. Let  $\mathcal{A}$  be a  $(n, k, k^2)$ -splitter obtained by Proposition 18. Let  $\mathcal{B}$  be a  $(k^2, k, \log k)$ -splitter obtained by Lemma 17. Let  $\mathcal{D}$  be a  $(k^2, k/\log k, q)$ -universal family obtained by Corollary 20. We construct  $\mathcal{F}$  as follows. For every function  $f_a$  in  $\mathcal{A}$ ,  $f_b$  in  $\mathcal{B}$ , and  $\log k$  functions  $g_1, g_2, \dots, g_{\log k}$  in  $\mathcal{D}$ , we construct a tuple  $f = (f_a, f_b, g_1, g_2, \dots, g_{\log k})$ , and add it to  $\mathcal{F}$ . We note here that  $g_1, g_2, \dots, g_{\log k}$  need not be different functions. For  $f \in \mathcal{F}$ , we define  $f : [n] \rightarrow [q]$  as follows. For  $x \in [n]$ , we have  $f(x) = g_r(f_b(f_a(x)))$ , where  $r = f_b(f_a(x))$ .

We first argue about the size of  $\mathcal{F}$  and the time needed to construct it. Notice that  $|\mathcal{F}| \leq |\mathcal{A}||\mathcal{B}||\mathcal{D}|^{\log k}$ . We know  $|\mathcal{A}| \leq k^{\mathcal{O}(1)} \log n$ ,  $|\mathcal{B}| \leq \mathcal{O}(k^{\mathcal{O}(\log k)})$  and  $|\mathcal{D}| \leq q^{k/\log k} k^{\mathcal{O}(k/\log k)}$  by Proposition 18, Lemma 17, and Corollary 20, respectively. This implies that  $|\mathcal{F}| \in \mathcal{O}(q^k \cdot k^{\mathcal{O}(\log k)} \cdot \log n)$ . Note that  $\mathcal{A}, \mathcal{B}, \mathcal{D}$  can be constructed in time  $\mathcal{O}(k^{\mathcal{O}(1)} n \log n)$ ,  $\mathcal{O}(k^{\mathcal{O}(\log k)})$ , and  $\mathcal{O}(q^k \cdot k^{\mathcal{O}(k/\log k)})$ , respectively. This implies that time required to construct  $\mathcal{F}$  is bounded by  $\mathcal{O}(q^k \cdot k^{\mathcal{O}(k)} \cdot n \log n)$ .

It remains to argue that  $\mathcal{F}$  has the desired properties. Consider  $S \subseteq [n]$  of size  $k$  and  $\phi : S \rightarrow [q]$ . We prove that there exists a function  $f \in \mathcal{F}$  such that  $f|_S \equiv \phi$ . By the definition of splitter, there exists  $f_a \in \mathcal{A}$  such that  $f_a$  evenly splits  $S$  (see Definition 16). Since  $|S| < k^2$ , for every  $y \in [k^2]$ ,  $|f_a^{-1}(y) \cap S|$  is either 0 or 1. Let  $Y = \{y_1, y_2, \dots, y_k\}$  be a subset of  $[k^2]$  such that  $y_1 < y_2 < \dots < y_k$  and  $|f_a^{-1}(y_i) \cap S| = 1$ , for all  $i \in [k]$ . For  $j = k/\log k$ , we mark every  $j^{\text{th}}$  element in set  $Y$  marking  $\log k - 1$  indices altogether. In other words, construct a subset  $Y'$  of  $Y$  of cardinality  $\log k - 1$  such that  $Y' = \{y_{1j}, y_{2j}, y_{3j} \dots, y_{(\log k - 1)j}\}$ . We use the set  $Y'$  to partition  $[k^2]$  in a way that every partition contains almost  $k/\log k$  many elements of  $Y$ . Let  $y_0 = 0$  and  $y_{(\log k)j} = k^2$  and define set  $Y_r = \{y \in Y \mid y_{r-1} < y \leq y_r\}$  for  $r \in [\log k]$ . Recall that a  $\mathcal{B}$  is  $(k^2, k, \log k)$ -splitter family obtained by Lemma 17. By construction, there exists a function  $f_b$  which corresponds to subset  $Y'$  of  $\log k - 1$  many indices. In other words, there is a function  $f_b$  such that  $f_b^{-1}(r)$  contains all the elements in  $Y_r$ , for each  $r$  in  $[\log k]$ . We note that size of  $f_b^{-1}(r)$  could be as large as  $k^2$ . Recall that  $\mathcal{D}$  is a  $(k^2, k/\log k, q)$ -universal family. Therefore, for every  $r \in [\log k]$  there exists  $g_r \in \mathcal{D}$  such that  $g_r|_{Y_r} \equiv \phi|_{Y_r}$ . Consider a function  $f = (f_a, f_b, g_1, g_2, \dots, g_{\log k})$  in  $\mathcal{F}$  where  $f_a, f_b$  and  $g_r$  satisfies the property mentioned above. The function  $f_a$  is bijective on  $S$  and  $f(S) = Y$ . The function  $f_b$  partitions  $Y$  into  $\log k$  many parts by mapping  $Y$  into  $Y_1, Y_2, \dots, Y_{\log k}$ . For each  $Y_r$  there exists a function  $g_r$  which gives the desired coloring of elements in  $Y_r$  and hence for the elements in  $S$ . Since we considering all possible combinations of  $f_a, f_b$  and  $\log k$  functions in  $\mathcal{D}$ , there exists a function  $f$  such that  $f|_S \equiv \phi$ , which proves the theorem. ◀

## 5 Non-existence of a Polynomial Kernel for $\mathbb{T}_\ell$ -Contraction

In this section, we show that  $\mathbb{T}_\ell$ -CONTRACTION does not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$ . We note that  $\mathbb{T}$ -CONTRACTION (TREE CONTRACTION) does not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$  [17]. We give a reduction from  $\mathbb{T}$ -CONTRACTION to  $\mathbb{T}_\ell$ -CONTRACTION as follows.

Let  $(G, k)$  be an instance of  $\mathbb{T}$ -CONTRACTION. We create an instance  $(G', k')$  of  $\mathbb{T}_\ell$ -CONTRACTION as follows. Initially, we have  $G = G'$ . Let  $v^*$  be an arbitrarily chosen vertex in  $V(G)$ . For each  $i \in [\ell]$ , we add a cycle  $(v^*, w_1^i, w_2^i, \dots, w_{k+1}^i)$  on  $k+2$  vertices to  $G'$ , which pairwise intersect at  $v^*$ , and we set  $k' = k$ . It is easy to see that  $(G, k)$  is a *yes* instance of  $\mathbb{T}$ -CONTRACTION if and only if  $(G', k')$  is a *yes* instance of  $\mathbb{T}_\ell$ -CONTRACTION.

► **Theorem 21.**  $\mathbb{T}_\ell$ -CONTRACTION does not admit a polynomial kernel unless  $\text{NP} \subseteq \text{coNP}/\text{poly}$ .

## 6 PSAKS for $\mathbb{T}_\ell$ -CONTRACTION

In this section, we design a PSAKS for  $\mathbb{T}_\ell$ -CONTRACTION, which complements the result that  $\mathbb{T}_\ell$ -CONTRACTION does not admit a polynomial kernel assuming  $\text{NP} \not\subseteq \text{coNP}/\text{poly}$  (Section 5).

Let  $(G, k)$  be an instance of  $\mathbb{T}_\ell$ -CONTRACTION. The algorithm starts by applying Reduction Rules 3.1 to 3.4 (if applicable, in that order). Next, we state the following lemma which will be useful in designing a reduction rule which will be employed for bounding the sizes of induced paths.

► **Lemma 22.** *Let  $(G, k)$  be an instance of  $\mathbb{T}_\ell$ -CONTRACTION and  $P = (u_0, u_1, \dots, u_q, u_{q+1})$  be a path in  $G$ , where  $q \geq k + 2$ , and for each  $i \in [q + 1]$  we have  $\deg(u_i) = 2$ . Then no minimal solution  $F$  to  $\mathbb{T}_\ell$ -CONTRACTION in  $(G, k)$  with  $|F| \leq k$  contains an edge incident to  $V(P) \setminus \{u_0, u_{q+1}\}$ .*

► **Reduction Rule 6.1.** *If  $G$  has a path  $P = (u_0, u_1, \dots, u_q, u_{q+1})$  such that  $q > k + 2$  and for all  $i \in [q]$ , we have  $\deg(u_i) = 2$ . Then contract the edge  $u_{q-1}u_q$ , i.e. the resulting instance is  $(G/\{u_{q-1}u_q\}, k)$ .*

Note that Reduction Rule 6.1 can be applied in polynomial time by searching for such a path (if it exists) in the subgraph induced on the vertices of degree 2 in  $G$ .

► **Lemma 23.** *Consider an instance  $(G, k)$  of  $\mathbb{T}_\ell$ -CONTRACTION on which Reduction Rule 6.1 is not applicable. If  $(G, k)$  is a yes instance of  $\mathbb{T}_\ell$ -CONTRACTION then  $G$  has a connected vertex cover of size at most  $2(k + 3)(k + 2\ell)$ .*

Before describing the next reduction rule, we define a partition of  $V(G)$  into the following sets. Let  $H = \{u \in V(G) \mid \deg(u) \geq 2(k + 3)(k + 2\ell) + 1\}$ ,  $I = \{v \in V(G) \setminus H \mid N(v) \subseteq H\}$ , and  $R = V(G) \setminus (H \cup I)$ . Vertices  $v, u$  are said to be *false twins* if  $N(v) = N(u)$ . We use Lemma 24 to reduce the number of vertices in  $I$  which have many false twins. Let  $G$  be  $k$ -contractible to a graph  $T$  in  $\mathbb{T}_\ell$  and  $\mathcal{W}$  be the  $T$ -witness structure of  $G$ .

► **Lemma 24.** *Consider sets  $X, U \subseteq V(G)$  such that  $U$  is an independent set in  $G$  and for all  $v \in U$  we have  $X \subseteq N(v)$ . If  $|U| \geq k + \ell + 2$  then there is a vertex  $t \in V(T)$  such that  $X \subseteq W(t)$ .*

► **Reduction Rule 6.2.** *If there is a vertex  $v \in I$  that has at least  $k + \ell + 2$  false twins in  $I$  then delete  $v$ , i.e. the resulting instance is  $(G - \{v\}, k)$ .*

For  $\alpha > 1$ , we let  $d = \lceil \frac{\alpha}{\alpha-1} \rceil$ . Next, we state our last reduction rule.

► **Reduction Rule 6.3.** *If there are vertices  $v_1, v_2, \dots, v_{k+\ell+2} \in I$  and  $h_1, h_2, \dots, h_d \in H$  such that for all  $i \in [k + \ell + 2]$ , we have  $\{h_1, \dots, h_d\} \subseteq N(v_i)$  then contract all edges in  $\tilde{E} = \{v_1 h_i \mid i \in [d]\}$ , and decrease  $k$  by  $d - 1$ . The resulting instance is  $(G/\tilde{E}, k - d + 1)$ .*

We note that the *lossy-ness* is introduced only in the Reduction Rule 6.3. We have determined that  $H' = \{h_1, h_2, \dots, h_d\}$  need to be in one witness bag but  $G[H']$  may not be connected. To simplify the graph, we introduce additional vertex  $v_1$  to the bag which contains  $H'$ . By doing this we are able to contract  $H' \cup \{v_1\}$  into a single vertex. In the following lemma, we argue that the number of extra edge contracted in this process is  $\alpha$  factor of the optimum solution.

► **Lemma 25.** *Let  $(G, k)$  be an instance of  $\mathbb{T}_\ell$ -CONTRACTION where none of the Reduction Rules 6.1 to 6.3 are applicable. If  $(G, k)$  is a yes of  $\mathbb{T}_\ell$ -CONTRACTION then  $|V(G)| \leq c[k(k + 2\ell)]^{d+1}$ , where  $c$  is some fixed constant.*

► **Theorem 26.**  *$\mathbb{T}_\ell$ -CONTRACTION admits a strict PSAKS, where the number of vertices is bounded by  $c[k(k + 2\ell)]^{\lceil \frac{\alpha}{\alpha-1} \rceil + 1}$ , where  $c$  is some fixed constant.*

## References

- 1 Akanksha Agrawal, Lawqueen Kanesh, Saket Saurabh, and Prafullkumar Tale. Paths to trees and cacti. In *CTAC*, pages 31–42, 2017.
- 2 Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Split contraction: The untold story. In *STACS*, volume 66 of *LIPICs*, pages 5:1–5:14, 2017.
- 3 Takao Asano and Tomio Hirata. Edge-Contraction Problems. *Journal of Computer and System Sciences*, 26(2):197–208, 1983.
- 4 Rémy Belmonte, Petr A. Golovach, Pim Hof, and Daniël Paulusma. Parameterized complexity of three edge contraction problems with degree constraints. *Acta Informatica*, 51(7):473–497, 2014.
- 5 Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
- 6 Leizhen Cai and Chengwei Guo. Contracting few edges to remove forbidden induced subgraphs. In *IPEC*, pages 97–109, 2013.
- 7 Marek Cygan. Deterministic parameterized connected vertex cover. In *Scandinavian Workshop on Algorithm Theory*, pages 95–106. Springer, 2012.
- 8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 9 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *FOCS*, pages 150–159, 2011.
- 10 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 11 Rod G. Downey and Michael R. Fellows. *Parameterized complexity*. Springer-Verlag, 1997.
- 12 Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized complexity*. Springer-Verlag, 2013.
- 13 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.
- 14 Petr A. Golovach, Pim van ’t Hof, and Daniel Paulusma. Obtaining planarity by contracting few edges. *Theoretical Computer Science*, 476:38–46, 2013.
- 15 Sylvain Guillemot and Dániel Marx. A faster FPT algorithm for bipartite contraction. *Inf. Process. Lett.*, 113(22–24):906–912, 2013.
- 16 Pinar Heggernes, Pim van ’t Hof, Daniel Lokshtanov, and Christophe Paul. Obtaining a bipartite graph by contracting few edges. *SIAM Journal on Discrete Mathematics*, 27(4):2143–2156, 2013.
- 17 Pinar Heggernes, Pim van ’t Hof, Benjamin Lévêque, Daniel Lokshtanov, and Christophe Paul. Contracting graphs to paths and trees. *Algorithmica*, 68(1):109–132, 2014.
- 18 Wenjun Li, Qilong Feng, Jianer Chen, and Shuai Hu. Improved kernel results for some FPT problems based on simple observations. *Theor. Comput. Sci.*, 657:20–27, 2017.
- 19 Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. On the hardness of eliminating small induced subgraphs by contracting edges. In *IPEC*, pages 243–254, 2013.
- 20 Moni Naor, Leonard J Schulman, and Aravind Srinivasan. Splitters and near-optimal derandomization. In *FOCS*, pages 182–191. IEEE, 1995.
- 21 Rolf Niedermeier. *Invitation to fixed-parameter algorithms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.
- 22 Toshimasa Watanabe, Tadashi Ae, and Akira Nakamura. On the removal of forbidden graphs by edge-deletion or by edge-contraction. *Discrete Applied Mathematics*, 3(2):151–153, 1981.
- 23 Toshimasa Watanabe, Tadashi Ae, and Akira Nakamura. On the NP-hardness of edge-deletion and-contraction problems. *Discrete Applied Mathematics*, 6(1):63–78, 1983.