

# On the Parameterized Complexity of Grid Contraction

**Saket Saurabh**

The Institute Of Mathematical Sciences, HBNI, Chennai, India  
University of Bergen, Bergen, Norway  
saket@imsc.res.in

**Uéverton dos Santos Souza**

Fluminense Federal Universidade, Niterói, Brazil  
ueverton@ic.uff.br

**Prafullkumar Tale**

Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany.  
prafullkumar.tale@mpi-inf.mpg.de

---

## Abstract

For a family of graphs  $\mathcal{G}$ , the  $\mathcal{G}$ -CONTRACTION problem takes as an input a graph  $G$  and an integer  $k$ , and the goal is to decide if there exists  $F \subseteq E(G)$  of size at most  $k$  such that  $G/F$  belongs to  $\mathcal{G}$ . Here,  $G/F$  is the graph obtained from  $G$  by contracting all the edges in  $F$ . In this article, we initiate the study of GRID CONTRACTION from the parameterized complexity point of view. We present a fixed parameter tractable algorithm, running in time  $c^k \cdot |V(G)|^{\mathcal{O}(1)}$ , for this problem. We complement this result by proving that unless ETH fails, there is no algorithm for GRID CONTRACTION with running time  $c^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$ . We also present a polynomial kernel for this problem.

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Fixed parameter tractability

**Keywords and phrases** Grid Contraction, FPT, Kernelization, Lower Bound

**Digital Object Identifier** 10.4230/LIPIcs.CVIT.2016.23

**Funding** *Saket Saurabh*: This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 819416), and Swarnajayanti Fellowship (No DST/SJF/MSA01/2017-18).



*Prafullkumar Tale*: This research is a part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement SYSTEMATICGRAPH (No. 725978).

## 1 Introduction

Graph modification problems are one of the central problems in graph theory that have received a lot of attention in theoretical computer science. Some of the important graph modification operations are vertex deletion, edge deletion, and edge contraction. For graph  $G$ , any graph that can be obtained from  $G$  by using these three types of modifications is called a *minor* of  $G$ . If only the first two types of modification operations are allowed then resulting graph is said to a *subgraph* of  $G$ . If the only third type of modification is allowed then the resulting graph is called a *contraction* of  $G$ .

For two positive integer  $r, q$ , the  $(r \times q)$ -grid is a graph in which every vertex is assigned a unique pair of the form  $(i, j)$  for  $1 \leq i \leq r$  and  $1 \leq j \leq q$ . A pair of vertices  $(i_1, j_1)$  and  $(i_2, j_2)$  are adjacent with each other if and only if  $|i_1 - i_2| + |j_1 - j_2| = 1$ . There has been considerable attention to the problem of obtaining a grid as a minor of the given graph. We find it surprising that the very closely related question of obtaining a grid as a contraction



© Saket Saurabh, Uéverton dos Santos Souza, and Prafullkumar Tale;  
licensed under Creative Commons License CC-BY

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:27

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

did not receive any attention. In this article, we initiate a study of this problem from the parameterized complexity point of view.

The *contraction* of edge  $uv$  in simple graph  $G$  deletes vertices  $u$  and  $v$  from  $G$ , and replaces them by a new vertex, which is made adjacent to vertices that were adjacent to either  $u$  or  $v$ . Note that the resulting graph does not contain self-loops and multiple edges. A graph  $G$  is said to be *contractible* to graph  $H$  if  $H$  can be obtained from  $G$  by edge contractions. Equivalently,  $G$  is contractible to  $H$  if  $V(G)$  can be partitioned into  $|V(H)|$  many connected sets, called witness sets, and these sets can be mapped to vertices in  $H$  such that adjacency between witness sets is consistent with their mapped vertices in  $H$ . If such a partition of  $V(G)$  exists then we call it  $H$ -witness structure of  $G$ . A graph  $G$  is said to be  $k$ -contractible to  $H$  if  $H$  can be obtained from  $G$  by  $k$  edge contractions. For a family of graphs  $\mathcal{G}$ , the  $\mathcal{G}$ -CONTRACTION problem takes as an input a graph  $G$  and an integer  $k$ , and the objective is to decide if  $G$  is  $k$ -contractible to a graph  $H$  in  $\mathcal{G}$ .

**Related Work :** Early papers of Watanabe et al. [21, 22], Asano and Hirata [3] showed  $\mathcal{G}$ -CONTRACTION is NP-Complete for various class of graphs like planar graphs, outer-planar graphs, series-parallel graphs, forests, chordal graphs. Brouwer and Veldman proved that it is NP-Complete even to determine whether a given graph can be contracted to a path of length four or not [5]. In the realm of parameterized complexity,  $\mathcal{G}$ -CONTRACTION has been studied with the parameter being the number of edges allowed to be contracted. It is known that  $\mathcal{G}$ -CONTRACTION admits an FPT algorithm when  $\mathcal{G}$  is set of paths [15], trees [15], cactus [18], cliques [6], planar graphs [12] and bipartite graphs [14, 13]. For a fixed integer  $d$ , let  $\mathcal{H}_{\geq d}$ ,  $\mathcal{H}_{\leq d}$  and  $\mathcal{H}_{=d}$  denote the set of graphs with minimum degree at least  $d$ , maximum degree at most  $d$ , and  $d$ -regular graphs, respectively. Golovach et al. [11] and Belmonte et al. [4] proved that  $\mathcal{G}$ -CONTRACTION admits an FPT algorithm when  $\mathcal{G} \in \{\mathcal{H}_{\geq d}, \mathcal{H}_{\leq d}, \mathcal{H}_{=d}\}$ . When  $\mathcal{G}$  is split graphs or chordal graphs, the  $\mathcal{G}$ -CONTRACTION is known to be W[1]-hard [2] and W[2]-hard [19, 6], respectively. To the best of our knowledge, it is known that  $\mathcal{G}$ -CONTRACTION admits a polynomial kernel only when  $\mathcal{G}$  is a set of paths [15] or set of paths or cycle i.e.  $\mathcal{H}_{\leq 2}$  [4]. It is known that  $\mathcal{G}$  does not admit a polynomial kernel, under standard complexity assumptions, when  $\mathcal{G}$  is set of trees [15], cactus [17], or cliques [6].

**Our Contribution :** In this article we study parameterized complexity of GRID CONTRACTION problem. We define the problem as follows.

GRID CONTRACTION

**Parameter:**  $k$

**Input:** Graph  $G$  and integer  $k$

**Question:** Is  $G$   $k$ -contractible to a grid?

To the best of our knowledge, the computation complexity of the problem is not known nor it is implied by the existing results regarding edge contraction problems. We prove that the problem is indeed NP-Complete (Theorem 27). We prove that there exists an FPT algorithm which given an instance  $(G, k)$  of GRID CONTRACTION runs in time  $4^{6k} \cdot |V(G)|^{\mathcal{O}(1)}$  and correctly concludes whether it is a YES instance or not (Theorem 22). We complement this result by proving that unless ETH fails there is no algorithm for GRID CONTRACTION with running time  $2^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$  (Theorem 27). We present a polynomial kernel with  $\mathcal{O}(k^4)$  vertices and edges for GRID CONTRACTION (Theorem 34).

**Our Methods :** Our FPT algorithm for GRID CONTRACTION is divided into two phases. In the first phase, we introduce a restricted version of GRID CONTRACTION problem called BOUNDED GRID CONTRACTION. In this problem, along with a graph  $G$  and an integer  $k$ , an input consists of an additional integer  $r$ . The objective is to determine whether graph  $G$  can be  $k$ -contracted to a grid with  $r$  rows. We present an FPT algorithm parameterized by  $(k + r)$

for this problem. This algorithm is inspired by the exact exponential algorithm for PATH CONTRACTION in [1]. It is easy to see that an instance  $(G, k)$  is a YES instance of GRID CONTRACTION if and only if  $(G, k, r)$  is a YES instance of BOUNDED GRID CONTRACTION for some  $r$  in  $\{1, 2, \dots, |V(G)|\}$ . In the second phase, given an instance  $(G, k)$  of GRID CONTRACTION we produce polynomially many instances of BOUNDED GRID CONTRACTION such that – (a) the input instance is a YES instance if and only if at least one of the produced instances is a YES instance and (b) for any produced instance, say  $(G', k', r)$ , we have  $k' = k$  and  $r \in \{1, 2, \dots, 2k + 5\}$ . We prove that all these instances can be produced in time polynomial in the size of the input. An FPT algorithm for GRID CONTRACTION is a direct consequence of these two results. We use techniques presented in the second phase to obtain a polynomial kernel for GRID CONTRACTION.

We present a brief overview of the FPT algorithm for BOUNDED GRID CONTRACTION. *Boundary vertices* of a subset  $S$  of  $V(G)$  are the vertices in  $S$  which are adjacent to at least one vertex in  $V(G) \setminus S$ . A subset  $S$  of  $V(G)$  is *nice* if both  $G[S]$ ,  $G - S$  are connected, and  $G[S]$  can be contracted to a  $(r \times q)$ -grid with all boundary vertices in  $S$  in an end-column for some integer  $q$ . In other words, a subset  $S$  of  $V(G)$  is nice if it is a union of witness sets appearing in first few columns in some grid witness structure of  $G$ . See Definition 11. The objective is to keep building a *special partial solution* for some nice subsets. In this special partial solution, all boundary vertices of a particular nice subset are contained in bags appearing in an end-column. This partial solution is then extended to the remaining graph. The central idea is – *for a nice subset  $S$  of graph  $G$ , if  $G[S]$  can be contracted to a grid such that all boundary vertices of  $S$  are in an end bag then how one contract  $G[S]$  is irrelevant.* This allows us to store one solution for  $G[S]$  and build a dynamic programming table nice subsets of vertices. The running time of such an algorithm depends on the following two quantities (i) the number of possible entries in the dynamic programming table, and (ii) time spent at each entry. We prove that *to bound both these quantities as a function of  $k$ , it is sufficient to know the size of neighborhood of  $S$  and the size of the union of witness sets in an end-column in a grid contraction of  $G[S]$  which contains all boundary vertices of  $S$ .*

In the second phase, we first check whether a given graph  $G$  can be  $k$ -contracted to a grid with  $r$  rows for  $r \in \{1, 2, \dots, 2k + 5\}$  using the algorithm mentioned in the previous paragraph. If for any value of  $r$  it returns YES then we can conclude that  $(G, k)$  is a YES instance of GRID CONTRACTION. Otherwise, we argue that there exists a *special separator*  $S$  in  $G$  which induces a  $(2 \times q)$  grid for some positive integer  $p$ . We prove that it is safe to contract  $q$  vertical edges in  $G[S]$ . Let  $G'$  be the graph obtained from  $G$  by contracting these parallel edges. Formally, we argue that  $G$  is  $k$ -contractible to a  $(r' \times q)$ -grid if and only if  $G'$  is  $k$ -contractible to  $((r' - 1) \times q)$ -grid. We keep repeating the process of finding a special separator and contracting parallel edges in it until one of the following things happens – (a) The resultant graph is  $k$ -contractible to a  $(r' \times q)$ -grid for some  $r' < 2k + 5$ . (b) The resultant graph does not contain a special separator. We argue that in Case (b), it is safe to conclude that  $(G, k)$  is a NO instance for GRID CONTRACTION.

**Organization of the paper :** We present some preliminary notations which will be used in rest of the paper in Section 2. We present a crucial combinatorial lemma in Section 3. As mentioned earlier, this algorithm is divided into two phases. We present the first and the second phase in Section 4 and 5, respectively. Section 5 also contains an FPT algorithm for GRID CONTRACTION. We prove that the dependency on the parameter in the running time of this algorithm is optimal, up to a constant factor, unless ETH fails in Section 6. In Section 7, we present a polynomial kernel for GRID CONTRACTION problem. We conclude the paper with an open question in Section 8.

## 2 Preliminaries

For a positive integer  $k$ ,  $[k]$  denotes the set  $\{1, 2, \dots, k\}$ .

### 2.1 Graph Theory

In this article, we consider simple graphs with a finite number of vertices. For an undirected graph  $G$ , sets  $V(G)$  and  $E(G)$  denote its set of vertices and edges respectively. Two vertices  $u, v$  in  $V(G)$  are said to be *adjacent* if there is an edge  $uv$  in  $E(G)$ . The neighborhood of a vertex  $v$ , denoted by  $N_G(v)$ , is the set of vertices adjacent to  $v$  and its degree  $d_G(v)$  is  $|N_G(v)|$ . The subscript in the notation for neighborhood and degree is omitted if the graph under consideration is clear. For a set of edges  $F$ , set  $V(F)$  denotes the collection of endpoints of edges in  $F$ . For a subset  $S$  of  $V(G)$ , we denote the graph obtained by deleting  $S$  from  $G$  by  $G - S$  and the subgraph of  $G$  induced on the set  $S$  by  $G[S]$ . For two subsets  $S_1, S_2$  of  $V(G)$ , we say  $S_1, S_2$  are adjacent if there exists an edge with one endpoint in  $S_1$  and other in  $S_2$ . For a subset  $S$  of  $V(G)$ , let  $\Phi(S)$  denotes set of vertices in  $S$  which are adjacent with at least one vertex outside  $S$ . Formally,  $\Phi(S) = \{s \in S \mid N(s) \setminus S \neq \emptyset\}$ . These are also called *boundary vertices* of  $S$ .

A *path*  $P = (v_1, \dots, v_l)$  is a sequence of distinct vertices where every consecutive pair of vertices is adjacent. For two vertices  $v_1, v_l$  in  $G$ ,  $\text{dist}(v_1, v_l)$  denotes the length of a shortest path between these two vertices. A graph is called *connected* if there is a path between every pair of distinct vertices. It is called *disconnected* otherwise. A set  $S$  of  $V(G)$  is said to be a *connected set* if  $G[S]$  is connected. For two vertices  $v_1, v_2$  in  $G$ , a set  $S$  is called  $(v_1-v_2)$ -*separator*, if any  $v_1-v_2$  paths intersects  $S$ . If a set is a  $(v_1-v_2)$ -separator as well as  $(v_3-v_4)$ -separator then we write it as  $\{(v_1-v_2), (v_3-v_4)\}$ -separator.

For two positive integer  $r, q$ , the  $(r \times q)$ -grid is a graph on  $r \cdot q$  vertices. The vertex set of this graph consists of all pairs of the form  $(i, j)$  for  $1 \leq i \leq r$  and  $1 \leq j \leq q$ . A pair of vertices  $(i_1, j_1)$  and  $(i_2, j_2)$  are adjacent with each other if and only if  $|i_1 - i_2| + |j_1 - j_2| = 1$ . We say that such a graph is a grid with  $r$  rows and  $q$  columns. It is called a  $(r \times q)$ -grid and is denoted by  $\boxplus_{r \times q}$ . We use  $\boxplus$  to denote a grid with an unspecified number of rows and columns. The vertices in grid  $\boxplus$  are denoted by  $\boxplus[i, j]$  or simply by  $[i, j]$ . Note that the grid with exactly one row is a path. To remove some corner cases, we consider grids that have at least two rows and two columns. Any grid contains exactly four vertices that have degree two. These vertices are called *corner vertices*. Let  $t_1 = [1, 1]$ ,  $t_2 = [1, q]$ ,  $t_3 = [r, q]$ , and  $t_4 = [r, 1]$  be the corner vertices in grid  $\boxplus_{r \times q}$ .

▷ **Observation 2.1.** If  $\hat{S}$  is a connected  $\{(t_1-t_4), (t_2-t_3)\}$ -separator in  $\boxplus_{r \times q}$  then its size is at least  $q$ . Moreover, if  $|\hat{S}| = q$  then it corresponds to a row in  $\boxplus_{r \times q}$ .

**Proof.** Without loss of generality, we assume that  $t_1 \equiv [1, 1]$ ,  $t_2 \equiv [1, q]$ ,  $t_3 \equiv [r, q]$ , and  $t_4 \equiv [r, 1]$ . Consider the  $(t_1 - t_4)$  path which contains vertices in the first column of the grid and the  $(t_2 - t_3)$  path which contains all vertices in the last column. Since  $\hat{S}$  separates  $t_1$  from  $t_4$  and  $t_2$  from  $t_3$ , it contains at least two vertices of the form  $[i, 1]$  and  $[i', q]$  for some  $i, i' \in \{1, 2, \dots, r\}$ . Since  $\hat{S}$  is connected then it contains a path connecting these two vertices. As any path connecting these two vertices contains at least  $q - 1 + |i - i'| + 1 \leq q$  vertices, the size of  $\hat{S}$  is at least  $q$ . If  $|\hat{S}| = q$  then  $|i - i'| = 0$ . This implies two endpoints of a row are contained in  $\hat{S}$ . Since  $\hat{S}$  is of size  $q$ , vertices that are present in  $\hat{S}$  are from one row. This proves the second part of the observation. ◀

## 2.2 Graph Contraction

The *contraction* of edge  $uv$  in  $G$  deletes vertices  $u$  and  $v$  from  $G$ , and adds a new vertex, which is made adjacent to vertices that were adjacent to either  $u$  or  $v$ . Notice that no self-loop or parallel edge is introduced in this process. The resulting graph is denoted by  $G/e$ . For a given graph  $G$  and edge  $e = uv$ , we formally define  $G/e$  in the following way:  $V(G/e) = (V(G) \cup \{w\}) \setminus \{u, v\}$  and  $E(G/e) = \{xy \mid x, y \in V(G) \setminus \{u, v\}, xy \in E(G)\} \cup \{wx \mid x \in N_G(u) \cup N_G(v)\}$ . Here,  $w$  is a new vertex which was not in  $V(G)$ . An edge contraction reduces the number of vertices in a graph by exactly one. Several edges might disappear due to one edge contraction. For a subset of edges  $F$  in  $G$ , graph  $G/F$  denotes the graph obtained from  $G$  by contracting each connected component in the sub-graph  $G' = (V(F), F)$  to a vertex.

► **Definition 1** (Graph Contraction). *A graph  $G$  is said to be contractible to graph  $H$  if there exists an onto function  $\psi : V(G) \rightarrow V(H)$  such that following properties hold.*

- For any vertex  $h$  in  $V(H)$ , graph  $G[W(h)]$  is connected and not empty, where set  $W(h) := \{v \in V(G) \mid \psi(v) = h\}$ .
- For any two vertices  $h, h'$  in  $V(H)$ , edge  $hh'$  is present in  $H$  if and only if there exists an edge in  $G$  with one endpoint in  $W(h)$  and another in  $W(h')$ .

We say graph  $G$  is contractible to  $H$  via mapping  $\psi$ . For a vertex  $h$  in  $H$ , set  $W(h)$  is called a *witness set* associated with/corresponding to  $h$ . We define *H-witness structure* of  $G$ , denoted by  $\mathcal{W}$ , as a collection of all witness set. Formally,  $\mathcal{W} = \{W(h) \mid h \in V(H)\}$ . A witness structure  $\mathcal{W}$  is a partition of vertices in  $G$ . If a *witness set* contains more than one vertex then we call it *big* witness-set, otherwise it is *small/singleton* witness set.

If graph  $G$  has a  $H$ -witness structure then graph  $H$  can be obtained from  $G$  by a series of edge contractions. For a fixed  $H$ -witness structure, let  $F$  be the union of spanning trees of all witness sets. By convention, the spanning tree of a singleton set is an empty set. To obtain graph  $H$  from  $G$ , it is necessary and sufficient to contract edges in  $F$ . We say graph  $G$  is *k-contractible* to  $H$  if cardinality of  $F$  is at most  $k$ . In other words,  $H$  can be obtained from  $G$  by at most  $k$  edge contractions. The following observations are immediate consequences of definitions.

▷ **Observation 2.2.** If graph  $G$  is  $k$ -contractible to graph  $H$  via mapping  $\psi$  then following statements are true.

1.  $|V(G)| \leq |V(H)| + k$ .
2. Any  $H$ -witness structure of  $G$  has at most  $k$  big witness sets.
3. For a fixed  $H$ -witness structure, the number of vertices in  $G$  which are contained in big witness sets is at most  $2k$ .
4. If  $S$  is a  $(x_1 - x_2)$ -separator in  $G$  then  $\psi(S)$  is a  $(\psi(x_1) - \psi(x_2))$ -separator in  $H$ .
5. If  $S$  is a separator in  $G$  such that there are at least two connected components of  $G \setminus S$  which has at least  $k + 1$  vertices, then  $\psi(S)$  is a separator in  $H$ .

**Proof.** The proof of (1), (2) and (3) follows directly from the definitions.

(4) Consider any  $(x_1 - x_2)$ -path  $P$  in  $G$ . Note that  $\psi(P)$  corresponds to a  $(\psi(x_1) - \psi(x_2))$ -path in  $H$  (with possible repetition of vertices). Since  $S$  is a  $(x_1 - x_2)$ -separator, every  $(x_1 - x_2)$ -path intersects with  $S$ . This implies that  $\psi(P)$  intersects  $\psi(S)$ . Since  $P$  is an arbitrary  $(x_1 - x_2)$ -path in  $G$ , we can conclude that every  $(\psi(x_1) - \psi(x_2))$ -path in  $H$  intersects  $\psi(S)$ . Hence  $\psi(S)$  is a  $(\psi(x_1) - \psi(x_2))$ -separator in  $H$ .

(5) Let  $C_a$  and  $C_b$  be two connected components of  $G \setminus S$  which has at least  $k + 1$  vertices. Since,  $G$  is  $k$ -contractible to  $H$ , there exists a vertex  $v_a$  in  $C_a$  (similarly,  $v_b$  in  $C_b$ ) such that  $\psi(v_a) \neq \psi(S)$  (similarly,  $\psi(v_b) \neq \psi(S)$ ). Hence,  $\psi(S)$  is a  $(\psi(v_a) - \psi(v_b))$ -separator in  $H$ .  $\blacktriangleleft$

## 2.3 Preliminary Result Regarding Grid Contraction

Suppose we are given a graph  $G$  with a mapping  $\psi$  such that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  via  $\psi$ . We define a notation of *partible row* in  $\boxplus_{r \times q}$  using mapping  $\psi$ . We argue that if  $\boxplus_{r \times q}$  contains a partible row then one can *un-contract* all vertices in this row to obtain a larger grid from  $G$ .

► **Definition 2** (Partible row). Consider a graph  $G$  which is  $k$ -contractible to  $\boxplus_{r \times q}$  via mapping  $\psi$ . The  $i_o^{th}$  row in  $\boxplus_{r \times q}$  is said to be partible if for every  $j$  in  $[q]$ , set  $\psi^{-1}([i_o, j])$  can be partitioned into non empty sets  $U_j$  and  $V_j$  which satisfy following properties:

- $G[U_j]$  and  $G[V_j]$  are connected.
- $U_j$  and  $V_{j'}$  are adjacent if and only if  $j = j'$ .
- $U_j$  and  $U_{j'}$  (similarly  $V_j$  and  $V_{j'}$ ) are adjacent if and only if  $|j - j'| = 1$ .
- Let  $U = \bigcup_{j \in [q]} U_j$  and  $V = \bigcup_{j \in [q]} V_j$ . If sets  $U, V$  are adjacent with  $C_b, C_f$ , respectively, then sets  $U, C_f$  (sets  $V, C_b$ ) are not adjacent.

Here,  $C_b := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i < i_o \text{ and } j \in [q]\}$ ;  $C_f := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i > i_o \text{ and } j \in [q]\}$ .

► **Lemma 3.** Consider a graph  $G$  which is  $k$ -contractible to  $\boxplus_{r \times q}$  via mapping  $\psi$ . If  $\boxplus_{r \times q}$  has a partible row then  $G$  is  $(k - q)$  contractible to  $\boxplus_{(r+1) \times q}$ .

**Proof.** Let  $i_o^{th}$  row be a partible row in  $\boxplus_{r \times q}$ . For  $j$  in  $[q]$ , let  $U_j$  and  $V_j$  be the partition of  $\psi^{-1}([i_o, j])$  which satisfy properties mentioned in Definition 2. Also, let  $U = \bigcup_{j \in [q]} U_j$  and  $V = \bigcup_{j \in [q]} V_j$ . Let set  $C_b$  (set  $C_f$ ) be the collection of vertices in  $G$  which are mapped to vertices in rows  $\{1, 2, \dots, i_o - 1\}$  (in rows  $\{i_o + 1, \dots, r\}$ ). Formally,  $C_b = \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i < i_o \text{ and } j \in [q]\}$  and  $C_f = \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i > i_o \text{ and } j \in [q]\}$ . Without loss of generality, we can assume that  $U, V$  are adjacent with  $C_b, C_f$ , respectively. Since set  $U$  (set  $V$ ) is a separator in  $G$ , sets  $U, C_f$  (sets  $V, C_b$ ) are not adjacent with each other. Note that  $\{C_b, U, V, C_f\}$  is a partition of  $V(G)$ . We define a function  $\phi : V(G) \rightarrow \boxplus_{(r+1) \times q}$  on  $V(G)$  as follows: for every  $x \in C_b$ ,  $\phi(x) = \psi(x)$ ; for every  $x \in C_f$ , if  $\psi(x) = [i, j]$  then  $\phi(x) = [i' + 1, j]$ ; for every  $x \in U_j$ ,  $\phi(x) = [i_o, j]$ ; and for every  $x \in V_j$ ,  $\phi(x) = [i_o + 1, j]$ . Since  $U_j, V_j$  are non-empty sets and  $\psi$  is an onto function,  $\phi$  is also an onto function. We argue that  $\phi$  satisfy both the properties mentioned in Definition 1.

For every vertex  $[i, j]$  in  $\boxplus_{r \times q}$ , set  $\psi^{-1}([i, j])$  is connected in  $G$ . Since  $G[U_j], G[V_j]$  are connected, for every  $[i, j]$  in  $\boxplus_{(r+1) \times q}$ , set  $\phi^{-1}([i, j])$  is connected in  $G$ . This proves the first property in Definition 1. To prove the second property, we argue that any two vertices, say  $[i_1, j_1]$  and  $[i_2, j_2]$ , in  $\boxplus_{(r+1) \times q}$  are adjacent with each other if and only if  $\phi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2])$  are adjacent with each other. Without loss of generality, we can assume that  $i_1 \leq i_2$ . Depending on the position of these two vertices in  $\boxplus_{(r+1) \times q}$ , we consider following five cases: (i)  $i_1, i_2 < i_o$ , (ii)  $i_1 < i_o$  and  $i_2 \in \{i_o, i_o + 1\}$ , (iii)  $\{i_1, i_2\} \subseteq \{i_o, i_o + 1\}$ , (iv)  $i_1 \in \{i_o, i_o + 1\}$  and  $i_2 > i_o + 1$ , and (v)  $i_1, i_2 > i_o + 1$ .

Consider Case (i). By definition of  $\phi$ , for  $i_1, i_2 < i_o$  we have  $\phi^{-1}([i_1, j_1]) = \psi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2]) = \psi^{-1}([i_2, j_2])$ . For  $i_1, i_2 < i_o$ , by the properties of  $\psi$ , there is an edge between  $[i_1, j_1]$  and  $[i_2, j_2]$  if and only if there is an edge between  $\psi^{-1}([i_1, j_1])$  and  $\psi^{-1}([i_2, j_2])$  in  $G$ . Hence we can conclude that for  $i_1, i_2 < i_o$  there is an edge between  $[i_1, j_1]$  and  $[i_2, j_2]$



if and only if there is an edge between  $\phi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2])$  in  $G$ . We can argue Case (v) by similar arguments on  $\psi^{-1}([i, j])$  and  $\phi^{-1}([i+1, j])$ . Consider Case (iii). Since  $U_j, V_{j'}$  are adjacent with each other if and only if  $j = j'$  and  $U_j, U_{j'}$  (similarly  $V_j, V_{j'}$ ) are adjacent with each other if and only if  $|j - j'| = 1$ . Hence, the second property is satisfied.

We now argue Case (ii). By the definition of  $\phi$ , we have  $\phi^{-1}([i_1, j_1]) = \psi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2]) \subseteq \psi^{-1}([i_2, j_2])$ . If there is an edge between  $[i_1, j_1]$  and  $[i_2, j_2]$  in  $\boxplus_{(r+1) \times q}$  then  $i_1 = i_o - 1$ ,  $i_2 = i_o$ , and  $j_1 = j_2 = j$  (say). By the property of  $\psi$ , there is an edge between  $\psi^{-1}([i_1, j])$  and  $\psi^{-1}([i_2, j])$ . Since  $C_b$  is adjacent with  $U$  and non adjacent with  $V$ , we know that  $\phi^{-1}([i_1, j]) (\subseteq C_b)$  is adjacent with  $U_j (\subseteq U)$  and non-adjacent with  $V_j (\subseteq V)$ . As  $U_j = \phi^{-1}([i_o, j]) = \phi^{-1}([i_2, j_2])$ , we can conclude that there exists an edge between  $\phi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2])$ . In reverse direction, suppose there exists an edge between  $\phi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2])$ . Since  $\phi^{-1}([i_1, j_1]) = \psi^{-1}([i_1, j_1])$  and  $\phi^{-1}([i_2, j_2]) \subseteq \psi^{-1}([i_2, j_2])$ , this implies there exists an edge between  $\psi^{-1}([i_1, j_1])$  and  $\psi^{-1}([i_2, j_2])$ . By the property of  $\psi$ , vertices  $[i_1, j_1]$  and  $[i_2, j_2]$  are adjacent with each other in  $\boxplus_{r \times q}$ . Since  $i_1 < i_o$  and  $i_2 \in \{i_o, i_o + 1\}$ , we can conclude that  $i_2 - i_1 = 1$  and  $j_1 = j_2$ . This implies  $[i_1, j_1], [i_2, j_2]$  are adjacent with each other in  $\boxplus_{(r+1) \times q}$ . We can argue Case (iv) using similar arguments.

Hence  $\phi : V(G) \rightarrow \boxplus_{(r+1) \times q}$  satisfies both the properties mentioned in Definition 1. This implies graph  $G$  is contractible to  $\boxplus_{(r+1) \times q}$  via  $\phi$ . As  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  and  $|V(\boxplus_{r \times q})| + q = |V(\boxplus_{(r+1) \times q})|$ , we can conclude that graph  $G$  is  $(k - q)$ -contractible to  $\boxplus_{(r+1) \times q}$ . ◀

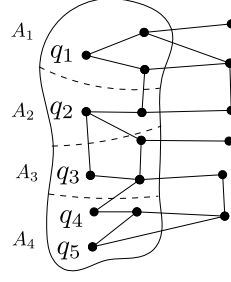
## 2.4 Parameterized Complexity

An instance of a parameterized problem comprises of an input  $I$ , which is an input of the classical instance of the problem and an integer  $k$ , which is called as the parameter. A problem  $\Pi$  is said to be *fixed-parameter tractable* or in **FPT** if given an instance  $(I, k)$  of  $\Pi$ , we can decide whether or not  $(I, k)$  is a YES instance of  $\Pi$  in time  $f(k) \cdot |I|^{\mathcal{O}(1)}$ . Here,  $f(\cdot)$  is some computable function whose value depends only on  $k$ . We say that two instances,  $(I, k)$  and  $(I', k')$ , of a parameterized problem  $\Pi$  are *equivalent* if  $(I, k) \in \Pi$  if and only if  $(I', k') \in \Pi$ . A *reduction rule*, for a parameterized problem  $\Pi$  is an algorithm that takes an instance  $(I, k)$  of  $\Pi$  as input and outputs an instance  $(I', k')$  of  $\Pi$  in time polynomial in  $|I|$  and  $k$ . If  $(I, k)$  and  $(I', k')$  are equivalent instances then we say the reduction rule is *safe*. A parameterized problem  $\Pi$  admits a kernel of size  $g(k)$  (or  $g(k)$ -kernel) if there is a polynomial time algorithm (called *kernelization algorithm*) which takes as an input  $(I, k)$ , and in time  $|I|^{\mathcal{O}(1)}$  returns an equivalent instance  $(I', k')$  of  $\Pi$  such that  $|I'| + k' \leq g(k)$ . Here,  $g(\cdot)$  is a computable function whose value depends only on  $k$ . For more details on parameterized complexity, we refer the reader to the books of Downey and Fellows [8], Flum and Grohe [9], Niedermeier [20], and the more recent books by Cygan et al. [7] and Fomin et al. [10].

## 3 Combinatorial Lemma

We introduce the notion of  $r$ -slabs which can be thought of as connected components with special properties. A  $r$ -slab is a connected set that can be partitioned into  $r$  connected subsets such that the adjacency between these parts and their neighbourhood follows a certain pattern. For an integer  $r$  and a set  $A$ , an *ordered  $r$ -partition* is a list of subsets of  $A$  whose union is  $A$ . We define  $r$ -slab as follows.

► **Definition 4 ( $r$ -Slab).** A  $r$ -slab in  $G$  is an ordered  $r$ -partition of a connected set  $A$ , say  $A_1, A_2, \dots, A_r$ , which satisfy following conditions.



■ **Figure 1** An example of a 4-slab. See Definition 4. For  $Q = \{q_1, q_2, q_3, q_4, q_5\}$  and its partition  $\mathcal{P}_4(Q) = \{\{q_1\}, \{q_2\}, \{q_3\}, \{q_4, q_5\}\}$ ,  $A$  is an  $(\mathcal{P}_4(Q), \alpha, \beta)$ -4-slab.

- For every  $i$  in  $[r]$ , set  $A_i$  is a non-empty set and  $G[A_i]$  is connected.
- For  $i \neq j$  in  $[r]$ , sets  $A_i, A_j$  are adjacent if and only if  $|i - j| = 1$ .
- For every  $i$  in  $[r]$ , define  $B_i = N(A_i) \setminus A$ . For  $i \neq j$  in  $[r]$ , sets  $B_i, B_j$  are mutually disjoint and if  $B_i$  and  $B_j$  are adjacent then  $|i - j| = 1$ .

We denote a  $r$ -slab by  $\langle A_1, A_2, \dots, A_r \rangle$ . For a  $r$ -slab  $\langle A_1, A_2, \dots, A_r \rangle$ , set  $A$  denotes union of all  $A_i$ s. We note that every connected subset of  $G$  is an 1-slab.

For positive integers  $\alpha, \beta$ , a connected set  $A$  in graph  $G$  is called an  $(\alpha, \beta)$ -connected set if  $|A| \leq \alpha$  and  $|N(A)| \leq \beta$ . For a non-empty set  $Q \subseteq V(G)$  a connected set  $A$  in  $G$  is a  $(Q)$ -connected set if  $Q \subseteq A$ . We generalize these notations for  $r$ -slab as follows.

► **Definition 5** ( $(\alpha, \beta)$ - $r$ -slab). For a graph  $G$  and integers  $\alpha, \beta$ , a  $r$ -slab  $\langle A_1, A_2, \dots, A_r \rangle$  is said to be an  $(\alpha, \beta)$ - $r$ -slab if  $|A| \leq \alpha$  and  $|N(A)| \leq \beta$ .

For a set  $Q$ , let  $\mathcal{P}_r(Q) = \{Q_1, Q_2, \dots, Q_r\}$  denotes its ordered  $r$ -partition. An ordered  $r$ -partition is said to be *valid* if for any two vertices  $u \in Q_i$  and  $v \in Q_j$ ,  $u, v$  are adjacent implies  $|i - j| \leq 1$ .

► **Definition 6** ( $\mathcal{P}_r(Q)$ - $r$ -slab). For a graph  $G$ , a subset  $Q$  of  $V(G)$  and its ordered valid partition  $\mathcal{P}_r(Q) = \{Q_1, Q_2, \dots, Q_r\}$ , a  $r$ -slab  $\langle A_1, A_2, \dots, A_r \rangle$  in  $G$  is said to be a  $\mathcal{P}_r(Q)$ - $r$ -slab if  $Q_i$  is a subset of  $A_i$  for every  $i$  in  $[r]$ .

See Figure 1 for an example. We combine properties mentioned in previous two definition to define specific types of  $r$ -slabs.

► **Definition 7** ( $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab). For a graph  $G$ , a non-empty subset  $Q$  of  $V(G)$ , its ordered valid partition  $\mathcal{P}_r(Q) = \{Q_1, Q_2, \dots, Q_r\}$ , and integers  $\alpha, \beta$ , a  $r$ -slab  $\langle A_1, A_2, \dots, A_r \rangle$  in  $G$  is a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab if it is an  $(\alpha, \beta)$ - $r$ -slab as well as a  $\mathcal{P}_r(Q)$ - $r$ -slab.

We mention following two observations which are direct consequences of the definition.

▷ **Observation 3.1.** Let  $\langle A_1, A_2, \dots, A_r \rangle$  be a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab in graph  $G$ . If a vertex  $v$  is in  $N(A)$  then  $\langle A_1, A_2, \dots, A_r \rangle$  is a  $(\mathcal{P}_r(Q), \alpha, \beta - 1)$ - $r$ -slab in graph  $G - \{v\}$ .

For a graph  $G$ , consider a vertex  $v$  and let  $G' = G - \{v\}$ . For a non-empty subset  $Q'$  of  $V(G')$ , its ordered partition  $\mathcal{P}_r(Q') = \{Q'_1, Q'_2, \dots, Q'_r\}$ , and integers  $\alpha, \beta$ , let  $\langle A'_1, A'_2, \dots, A'_r \rangle$  be a  $(\mathcal{P}_r(Q'), \alpha, \beta)$ - $r$ -slab in  $G'$ .

▷ **Observation 3.2.** If vertex  $v$  satisfy following two properties then  $\langle A'_1, A'_2, \dots, A'_r \rangle$  is a  $(\mathcal{P}_r(Q'), \alpha, \beta + 1)$ - $r$ -slab in  $G$ .



- Vertex  $v$  is adjacent with exactly one part, say  $A'_i$ , of the  $r$ -slab
- For any vertex  $u$  in  $N'_G(A'_j) \setminus A'$ , if  $u$  and  $v$  are adjacent in  $G$  then  $|i - j| \leq 1$ .

Definition 7 generalizes the notation of  $(Q, \alpha, \beta)$ -connected set defined in [1]. In the same paper, authors proved that there is an algorithm that given a graph  $G$  on  $n$  vertices, a non-empty set  $Q \subseteq V(G)$ , and integers  $\alpha, \beta$ , enumerates all  $(Q, \alpha, \beta)$ -connected sets in  $G$  in time  $2^{\alpha - |Q| + \beta} \cdot n^{O(1)}$ . We present similar combinatorial lemma for  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs.

► **Lemma 8.** *There is an algorithm that given a graph  $G$  on  $n$  vertices, a non-empty set  $Q \subseteq V(G)$ , its ordered partition  $\mathcal{P}_r(Q) = \{Q_1, Q_2, \dots, Q_r\}$ , and integers  $\alpha, \beta$ , enumerates all  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$  in time  $4^{\alpha - |Q| + \beta} \cdot n^{O(1)}$ .*

**Proof.** Let  $N(Q) = \{v_1, v_2, \dots, v_p\}$ . Arbitrarily fix a vertex  $v_l$  in  $N(Q)$ . We partition  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$  based on whether  $v_l$  is contained in it or not. In later case, such  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab is also a  $(\mathcal{P}_r(Q), \alpha, \beta - 1)$ - $r$ -slab in  $G - \{v\}$ . We now consider the first case. Let  $i$  be the smallest integer in  $[r]$  such that  $v_l$  is adjacent with  $Q_i$ . Note that, by definition, if  $v_l$  is present in a  $\mathcal{P}_r(Q)$ - $r$ -slab then it can be part of either  $A_{i-1}$ ,  $A_i$  or  $A_{i+1}$ . We encode this fact by moving  $v_l$  to either  $Q_{i-1}$ ,  $Q_i$  or  $Q_{i+1}$ . Let  $\mathcal{P}_r^{i-1}(Q \cup \{v_l\})$ ,  $\mathcal{P}_r^i(Q \cup \{v_l\})$  and  $\mathcal{P}_r^{i+1}(Q \cup \{v_l\})$  be  $r$ -partitions of  $Q \cup \{v_l\}$  obtained from  $\mathcal{P}_r(Q)$  by adding  $v_l$  to set  $Q_{i-1}$ ,  $Q_i$  and  $Q_{i+1}$ , respectively. Formally, these three sets are defined as follows.

- $\mathcal{P}_r^{i-1}(Q \cup \{v_l\}) := \{Q_1, \dots, Q_{i-1} \cup \{v_l\}, Q_i, Q_{i+1}, \dots, Q_r\}$
- $\mathcal{P}_r^i(Q \cup \{v_l\}) := \{Q_1, \dots, Q_{i-1}, Q_i \cup \{v_l\}, Q_{i+1}, \dots, Q_r\}$
- $\mathcal{P}_r^{i+1}(Q \cup \{v_l\}) := \{Q_1, \dots, Q_{i-1}, Q_i, Q_{i+1} \cup \{v_l\}, \dots, Q_r\}$

**Algorithm :** We present a recursive enumeration algorithm which takes  $(G, \mathcal{P}_r(Q), \alpha, \beta)$  as an input and outputs a set, say  $\mathcal{A}$ , of all  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab in  $G$ . The algorithm initializes  $\mathcal{A}$  to an empty set. The algorithm returns  $\mathcal{A}$  if one of the following statements is true: (i)  $\mathcal{P}_r(Q)$  is not a valid partition of  $Q$ , (ii)  $\alpha - |Q| < 0$  or  $\beta < 0$ , (iii) there is a vertex  $v_l$  in  $N(Q)$  which is adjacent with  $Q_i$  and  $Q_j$  for some  $i, j$  in  $[r]$  such that  $|i - j| \geq 2$ . If  $\alpha - |Q| + \beta = 0$ , the the algorithm checks if  $\mathcal{P}_r(Q)$  is a  $(\mathcal{P}_r(Q), 0, 0)$ - $r$ -slabs in  $G$ . If it is the case then the algorithm returns singleton set containing  $\mathcal{P}_r(Q)$  otherwise it returns an empty set. If there is a vertex  $v_l$  in  $N(Q)$  which is adjacent with  $Q_{i-1}$ ,  $Q_i$  and  $Q_{i+1}$  for some  $i$  in  $[r]$  then the algorithm calls itself on instance  $(G, \mathcal{P}_r^i(Q \cup \{v_l\}), \alpha, \beta)$  where  $\mathcal{P}_r^i(Q \cup \{v_l\})$  is  $r$ -partition as defined above. It returns the set obtained on this recursive call as the output. If there are no such vertices in  $N(Q)$ , then for some  $l \in \{1, \dots, |N(Q)|\}$ , the algorithm creates four instances viz  $(G - \{v_l\}, \mathcal{P}_r(Q), \alpha, \beta - 1)$  and  $(G, \mathcal{P}_r^{i_0}(Q \cup \{v_l\}), \alpha, \beta)$  for  $i_0 \in \{i - 1, i, i + 1\}$ . The algorithm calls itself recursively on these four instances. Let  $\mathcal{A}_l^v$ ,  $\mathcal{A}_l^{i-1}$ ,  $\mathcal{A}_l^i$ , and  $\mathcal{A}_{i+1}$  be the set returned, respectively, by the recursive call of the algorithm. The algorithm adds all elements in  $\mathcal{A}_l^{i-1} \cup \mathcal{A}_l^i \cup \mathcal{A}_l^{i+1}$  to  $\mathcal{A}$ . For every  $(\mathcal{P}_r(Q), \alpha, \beta - 1)$ - $r$ -slabs  $\langle A'_1, A'_2, \dots, A'_r \rangle$  in  $\mathcal{A}_l^v$ , the algorithm checks whether it is a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$  using Observation 3.2. If it is indeed a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$  then it adds it to  $\mathcal{A}$ . The algorithm returns  $\mathcal{A}$  at the end of this process.

We now argue the correctness of the algorithm. For every input instance  $(G, \mathcal{P}_r(Q), \alpha, \beta)$  we define its measure as  $\mu((G, \mathcal{P}_r(Q), \alpha, \beta)) = \alpha - |Q| + \beta$ . We proceed by the induction hypothesis that the algorithm is correct on any input whose measure is strictly less than  $\alpha - |Q| + \beta$ . Consider the base cases  $\alpha - |Q| + \beta = 0$ . In this case, the only possible  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab is  $\mathcal{P}_r(Q)$ . The algorithm checks this and returns the correct answer accordingly. We consider the case when  $\alpha - |Q| + \beta \geq 1$ . Every  $(\mathcal{P}_r(Q \cup \{v_l\}), \alpha, \beta)$ - $r$ -slab is also a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab. The algorithm adds a  $r$ -slab in  $\mathcal{A}_l^v$  to  $\mathcal{A}$  only if it is a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$ . Hence the algorithm returns a set of  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$ . In remaining part we argue that every  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs is enumerated by the algorithm.

By Definition 4, no vertex in closed neighborhood of a  $r$ -slab can be adjacent to two non-adjacent parts of a  $r$ -slab. Hence, if there is a vertex  $v_l$  in  $N(Q)$  which is adjacent with  $Q_i$  and  $Q_j$  for some  $i, j$  in  $[r]$  such that  $|i - j| \geq 2$  then the algorithm correctly returns an empty set. Suppose there exists a vertex  $v$  in  $N(Q)$  which is adjacent with  $Q_{i-1}, Q_i$  and  $Q_{i+1}$  for some  $i$  in  $[r]$ . By Definition 4, any  $r$ -slab containing  $\mathcal{P}_r(Q)$  must contains  $v$  in it. In this case, the number of  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab is same as the number of  $(\mathcal{P}_r^i(Q \cup \{v\}), \alpha, \beta)$ - $r$ -slab where  $\mathcal{P}_r^i(Q \cup \{v\})$  is the  $r$ -partition of  $Q \cup \{v\}$  obtained from  $\mathcal{P}_r(Q)$  by adding  $v$  to  $Q_i$ . The measure for input instance  $(G, \mathcal{P}_r^i(Q \cup \{v\}), \alpha, \beta)$  is strictly smaller than  $\alpha - |Q| + \beta$ . Hence by induction hypothesis, the algorithm correctly computes all  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab.

Consider the case when there is no vertex which is adjacent with  $Q_{i-1}, Q_i$  and  $Q_{i+1}$  for any  $i$  in  $[r]$ . Let  $v_l$  be a vertex in  $N(Q)$  and there is an integer  $i$  in  $[p]$  such that  $i$  is the smallest integer, and  $v_l$  is adjacent with  $Q_i$ . As mentioned earlier, either  $v_l$  is a part of  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab or not. In first case, by Definition 4,  $v_l$  can be part of  $A_{i-1}, A_i$  or  $A_{i+1}$  in any  $\mathcal{P}_r(Q)$ - $r$ -slab. The measure of input instance  $(G, \mathcal{P}_r^{i_0}(Q \cup \{v\}), \alpha, \beta)$  is  $\alpha - |Q| + \beta - 1$ . Hence by induction hypothesis, the algorithm correctly enumerates all  $(\mathcal{P}_r^{i_0}(Q \cup \{v\}), \alpha, \beta)$ - $r$ -slabs in  $G_l$ . Consider a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab  $\langle A_1, A_2, \dots, A_r \rangle$  in  $G$  which does not contain  $v_l$ . By Observation 3.1,  $\langle A_1, A_2, \dots, A_r \rangle$  is a  $(\mathcal{P}_r(Q), \alpha, \beta - 1)$ - $r$ -slab in  $G - \{v\}$ . By induction hypothesis, the algorithm correctly computes all  $(\mathcal{P}_r(Q), \alpha, \beta - 1)$ - $r$ -slabs in  $G - \{v\}$ . Since  $\langle A_1, A_2, \dots, A_r \rangle$  is a  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slab in  $G$ , vertex  $v_l$  satisfy both the properties mentioned in Observation 3.2. Hence algorithm adds  $\langle A_1, A_2, \dots, A_r \rangle$  to the set  $\mathcal{A}_l$ . Hence, we can conclude that the algorithm correctly enumerates all  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in  $G$ .

Using the induction hypothesis that the algorithm correctly outputs the set of all  $(\mathcal{P}_r(Q), \alpha, \beta)$ - $r$ -slabs in time  $4^{\alpha - |Q| + \beta} \cdot n^{\mathcal{O}(1)}$ , the running time of the algorithm follows. This concludes the proof of the lemma.  $\blacktriangleleft$

We use following corollary of Lemma 8.

► **Corollary 9.** *There is an algorithm that given a graph  $G$  on  $n$  vertices and integers  $\alpha, \beta$ , enumerates all  $(\alpha, \beta)$ - $r$ -slab in  $G$  in time  $4^{\alpha + \beta} \cdot n^{\mathcal{O}(1)}$ .*

## 4 An FPT algorithm for Bounded Grid Contraction

In this section, we present an FPT algorithm for BOUNDED GRID CONTRACTION. We formally define the problem as follows.

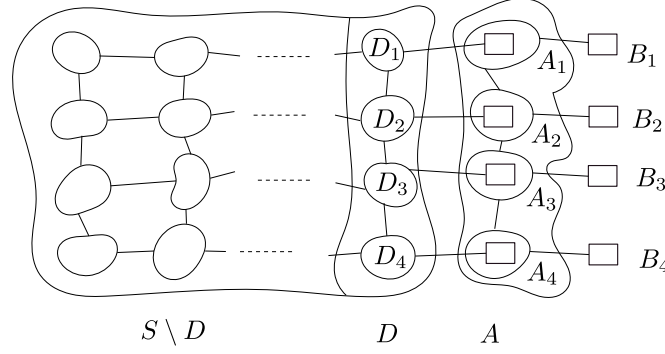
BOUNDED GRID CONTRACTION

**Parameter:**  $k, r$

**Input:** Graph  $G$  and integers  $k, r$

**Question:** Is  $G$   $k$ -contractible to a grid with  $r$  rows?

We start with a definition of nice subsets mentioned in the Introduction section. As mentioned before, vertices of a nice subset correspond to witness sets in the first few columns of a grid-witness structure of the input graph. Hence boundary vertices of a nice set correspond to witness sets in some column of a grid. Note that we are interested in the grids that have exactly  $r$ -rows. Hence, we use the notation of  $r$ -slab defined in previous section to formally define nice sets. Consider a  $r$ -slab  $\langle D_1, D_2, \dots, D_r \rangle$  which corresponds to a column in some grid that can be obtained from the input graph with at most  $k$  edge contraction. By Observation 2.2, an edge contraction reduces the number of vertices by exactly one. As there are  $3r$  many vertices in three adjacent rows in a grid, the size of a closed neighborhood of  $D$  in  $G$  is at most  $k + 3r$ . Thus, we can focus our attention on  $r$ -slabs with the bounded closed neighborhood. We define  $k$ -potential  $r$ -slabs as follows.



■ **Figure 2** All sets with smooth (non-rectangular) boundary are connected. Set  $A$  is a possible extension of nice subset  $S$ . In other words,  $A$  is an element in  $\mathcal{A}_{|A|,|B|}[(S, \mathcal{P}_r(D))]$ . See paragraph before Lemma 13.

► **Definition 10** (*k*-Potential *r*-Slab). For a given graph  $G$  and integers  $k, r$ , a *r*-slab  $\langle D_1, D_2, \dots, D_r \rangle$  is said to be a *k*-potential *r*-slab of  $G$  if it satisfies following two conditions:

- $|D| + |N(D)| \leq k + 3r$ ; and
- $G - D$  has at most two connected components.

Here,  $D = D_1 \cup D_2 \cup \dots \cup D_r$ .

► **Definition 11** (Nice Subset). A subset  $S$  of  $V(G)$  is said to be a nice subset of  $G$  if there exists a *k*-potential *r*-slab, say  $\langle D_1, D_2, \dots, D_r \rangle$ , such that  $D$  is a subset of  $S$  and  $G[S \setminus D]$  is one of the connected components of  $G - D$ . We say that *r*-slab  $\langle D_1, D_2, \dots, D_r \rangle$  is responsible for nice subset  $S$ .

Since  $\langle D_1, D_2, \dots, D_r \rangle$  is a *k*-potential *r*-slab, both  $G[S]$  and  $G - S$  are connected. There may be more than one *k*-potential *r*-slabs responsible for a nice subset. We define a pair of nice sets and *k*-potential *r*-slabs responsible for it.

► **Definition 12** (Valid Tuple). A tuple  $(S, \mathcal{P}_r(D))$  is called a valid tuple if  $S$  is a nice subset and  $\mathcal{P}_r(D) \equiv \langle D_1, D_2, \dots, D_r \rangle$  is a *k*-potential *r*-slab responsible for it.

Let  $\mathcal{V}_k$  be the set of all valid tuples. For a valid tuple  $(S, \mathcal{P}_r(D))$  in  $\mathcal{V}_k$ , we define a collection of *k*-potential *r*-slabs which is denoted by  $\mathcal{A}[(S, \mathcal{P}_r(D))]$ . This set can be thought of as a collection of “potential column extenders” for  $S$ . See Figure 2. In other words, we can append a *k*-potential-*r*-slab in  $\mathcal{A}[(S, \mathcal{P}_r(D))]$  to get a grid witness structure of a larger graphs containing  $S$ . Let  $\mathcal{P}_r(A)$  be a *k*-potential-*r*-slab in  $\mathcal{A}[(S, \mathcal{P}_r(D))]$ . Intuitively speaking,  $\mathcal{P}_r(A)$  is the “new” column to be “appended” to a grid witness structure of  $G[S]$ , to obtain a grid witness structure for  $G[S \cup A]$ . Hence if  $G[S]$  can be  $k'$ -contracted to a grid then  $G[S \cup A]$  can be  $k' + (|A| - r)$ -contracted to a grid. For improved analysis, we concentrate on subset  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]$  of  $\mathcal{A}[(S, \mathcal{P}_r(D))]$  defined for integers  $a, b$ . The set  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]$  is a collection of *k*-potential *r*-slabs of size at most  $a$  which have at most  $b$  neighbors outside  $S$ . We impose additional condition that  $a + b + |D|$  is at most  $k + 3r$  for improved analysis. Formally,  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))] = \{ \langle A_1, A_2, \dots, A_r \rangle \mid |A| \leq a, |N(A) \setminus S| \leq b, \text{ where } A = A_1 \cup A_2 \cup \dots \cup A_r \text{ and for every } D_i \text{ in } \mathcal{P}_r(D), (N(D_i) \setminus S) \subseteq A_i, \text{ and } a + b + |D| \leq k + 3r \}$ .

**Algorithm :** The algorithm takes a graph  $G$  on  $n$  vertices and integers  $k, r$  as input and outputs either **True** or **False**. The algorithm constructs a dynamic programming table  $\Gamma$  in

which there is an entry corresponding to every index  $[(S, \mathcal{P}_r(D)); k']$  where  $(S, \mathcal{P}_r(D))$  is a valid tuple in  $\mathcal{V}_k$  and  $k'$  is an integer in  $\{0\} \cup [k]$ . It initialize values corresponding to all entries to **False**.

(*for-loop Initialization*) For a tuple  $(S, \mathcal{P}_r(D)) \in \mathcal{V}_k$  such that  $S = D$  and  $k' \geq |S| - r = |D| - r$ , the algorithm sets  $\Gamma[(S, \mathcal{P}_r(D)); k'] = \mathbf{True}$ .

(*for-loop Table*) The algorithm processes indices in the table in chronologically increasing order. It first checks the size of  $S$ , then the size of  $D$ , followed by  $k$ . Ties are broken arbitrarily. At table index  $[(S, \mathcal{P}_r(D)); k']$ , if  $\Gamma[(S, \mathcal{P}_r(D)); k']$  is **False** then the algorithm continues to next tuple. If  $\Gamma[(S, \mathcal{P}_r(D)); k']$  is **True** then it runs the following for-loop at this index.

(*for-loop at Index*) The algorithm computes the set  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]$  for every pair of integers  $a (\geq r), b (\geq 0)$  which satisfy following properties (1)  $a + b + |D| \leq k + 3r$ , (2)  $k' + a - r \leq k$ , and (3)  $|N(S)| \leq a$ . For every  $k$ -potential  $r$ -slab  $\mathcal{P}_r(A)$  in  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]$ , the algorithm sets  $\Gamma[(S \cup A, \mathcal{P}_r(A)); k_1]$  to **True** for every  $k_1 \geq k' + (a - r)$ .

If  $\Gamma[(V(G), \mathcal{P}_r(D)); k']$  is set to **True** for some  $\mathcal{P}_r(D)$  and  $k'$  then the algorithm returns **True** otherwise it returns **False**. This completes the description of the algorithm.

Recall that for a given connected subset  $S$  of  $V(G)$ ,  $\Phi(S)$  denotes its boundary vertices i.e. set of vertices in  $S$  which are adjacent with at least one vertex outside  $S$ .

► **Lemma 13.** *For every tuple  $(S, \mathcal{P}_r(D))$  in  $\mathcal{V}_k$  and integer  $k'$  in  $\{0\} \cup [k]$ , the algorithm assign  $\Gamma[(S, \mathcal{P}_r(D)); k'] = \mathbf{True}$  if and only if  $k' + |N(S)| - r \leq k$  and there is a  $(r \times q)$ -grid witness structure of  $G[S]$ , for some integer  $q$ , such that  $\mathcal{P}_r(D)$  is collection of witness sets in an end-column and  $\Phi(S)$  is in  $D$ .*

**Proof.** We prove the lemma by induction on  $|S| + k'$  for indices  $((S, \mathcal{P}_r(D)); k')$  in the dynamic programming table. For the induction hypothesis, we assume that for a positive integer  $z$  the algorithm computes  $\Gamma[(S, \mathcal{P}_r(D)); k']$  correctly for each  $(S, \mathcal{P}_r(D))$  in  $\mathcal{V}_k$  and  $k'$  in  $0 \cup [k]$  for which  $|S| + k' \leq z$ .

Consider the base case when  $|S| = |D| = r$  and  $k' = 0$ . Since  $D \subseteq S$ , we have  $S = D$ . This implies  $\mathcal{P}_r(S) = \mathcal{P}_r(D)$  is a  $r$ -slab. Any connected subset of a graph can be contracted to a vertex by contracting a spanning tree. Hence,  $G[S]$  can be contracted to a  $(r \times 1)$ -grid by contracting  $|D| - r$  many edges. This implies that the values assigned by the algorithm in (*for-loop Initialization*) are correct. We note that once the algorithm sets a particular value to **True**, it does not change it afterwards.

Assuming induction hypothesis, we now argue that the computation of  $\Gamma[\cdot]$  for indices of the form  $[(S_1, \mathcal{P}_r(D_1)); k_1]$  where  $|S_1| + k_1 = z + 1$  are correct. Note that if  $[(S_1, \mathcal{P}_r(D_1)); k_1]$  is an entry in the table then  $(S_1, \mathcal{P}_r(D_1))$  is a valid tuple in  $\mathcal{V}_k$  and  $k_1$  is an integer in the set  $\{0\} \cup [k]$ .

( $\Rightarrow$ ) Assume that  $G[S_1]$  is  $k_1$ -contractible to a  $(r \times q)$ -grid such that all vertices in  $\Phi(S_1)$  are in an end-column  $\mathcal{P}_r(D_1)$  and  $k_1 + |N(S_1)| - r \leq k$ . We argue that the algorithm sets  $\Gamma[(S_1, \mathcal{P}_r(D_1)); k_1]$  to **True**. Let  $G[S_1]$  be  $k_1$ -contractible to a  $(r \times q)$ -grid. If  $q = 1$  then  $D_1 = S_1$  and in this case algorithm correctly computes  $\Gamma[(S_1, \mathcal{P}_r(D_1)); k_1]$ . Consider the case when  $q \geq 2$ . Let  $\mathcal{W} = \{W_{ij} \mid (i, j) \in [r] \times [q]\}$  be a  $(r \times q)$ -grid structure of  $G$  such that  $\mathcal{P}_r(D)$  is collection of witness sets in an end-column and  $\Phi(S_1)$  is a subset of  $D$ . Define  $W_j^c$  as union of all witness sets in column  $j$ . Formally,  $W_j^c = \bigcup_{i=1}^r W_{ij}$ . Hence,  $\mathcal{W} = W_1^c \cup W_2^c \cup \dots \cup W_{q-1}^c \cup W_q^c$  and  $\mathcal{P}_r(D) = W_q^c$ . Consider set  $S_0 = W_1^c \cup W_2^c \cup \dots \cup W_{q-1}^c$ . Since  $q \geq 2$ ,  $S_0$  is a non-empty set. Let  $k_0 = k_1 - (|W_q^c| - r)$ . We argue that  $[(S_0, W_{q-1}^c); k_0]$  is an index in the table and  $|S_0| + k_0 \leq z$ . As  $\mathcal{W}$  is a  $k_1$ -grid witness structure,  $|W_q^c| - r \leq k_1$  and hence  $k_0$  is a non-negative integer. Since  $G[W_q^c]$  is a connected graph,  $G - W_{q-1}^c$  has

exactly two connected components viz  $G[W_1^c \cup \dots \cup W_{q-2}^c]$  and the component containing  $W_q^c$ . As  $\mathcal{W}$  is a  $k_1$ -grid witness structure,  $|W_{q-2}^c| + |W_{q-1}^c| + |W_q^c| \leq k_1 + 3r \leq k + 3r$  and  $N(W_{q-1}^c) \subseteq W_{q-2}^c \cup W_q^c$ . (We note that  $W_{q-2}^c$  may not exist but this does not change the argument. For the sake of clarity, we do not consider this as separate case.) Since  $|W_{q-1}^c| + |N(W_{q-1}^c)| \leq k + 3r$  and  $G - W_{q-1}^c$  has at most two connected components,  $W_{q-1}^c$  is a  $k$ -potential  $r$ -slab. Note that  $\langle W_{1j}, W_{2j}, \dots, W_{rj} \rangle$  is the  $r$ -partition of  $k$ -potential  $r$ -slab  $W_{q-1}^c$ . Hence  $(S_0, W_{q-1}^c)$  is a tuple in  $\mathcal{V}_k$  and  $((S_0, W_{q-1}^c); k_0)$  is an index in the table. Since  $W_q^c$  is not an empty set,  $|S_0| + k_0 \leq |S_1| - |W_q^c| + k_1 - (|W_q^c| - r) \leq z + 1 + r - 2|W_q^c|$  as  $|S_1| + k_1 = z + 1$ . Since  $|W_q^c| \geq r \geq 1$ , we conclude  $|S_0| + k_0 \leq z$ . Note that  $\mathcal{W} \setminus \{W_q^c\}$  is a  $(k_1 - |W_q^c| + r)$ -grid witness structure for  $G[S_0]$ . This implies that  $G[S_0]$  is  $k_0$ -contractible to a grid with  $W_{q-1}^c$  as collection of bags in an end-column and  $k_0 + |N(S_0)| - r \leq k_1 \leq k$ . Moreover,  $S_0 = S_1 \setminus W_q^c$ ,  $\Phi(S_0)$  is contained in  $W_{q-1}^c$ . By the induction hypothesis, the algorithm has correctly set  $\Gamma[(S_0, W_{q-1}^c); k_0]$  to **True**. Let  $x_0 = |W_{q-1}^c|$ ,  $a = |W_q^c|$  and  $b = |W_q^c \setminus N(S_0)| = |N(S_1)|$ . We first claim that  $x_0 + a + b \leq k + 3r$ . Note that  $|W_{q-1}^c| + |W_q^c| \leq k_1 + 2r$  and  $k_1 + b \leq k + r$ . Hence  $|W_{q-1}^c| + |W_q^c| + b = x_0 + a + b \leq k + 3r$ . At index  $[(S_0, W_{q-1}^c); k_0]$ , the algorithm computes  $\mathcal{A}_{a,b}[(S_0, W_{q-1}^c)]$ . Clearly,  $W_q^c$  is one of the sets in  $\mathcal{A}_{a,b}[(S_0, W_{q-1}^c)]$  as for every  $i$  in  $[r]$ ,  $N(W_{i,q-1}) \setminus S_0$  is contained in  $W_{iq}$  and  $G[W_{iq}]$  is a connected graph. Hence the algorithm sets  $\Gamma[(S_1, W_q^c), k_1] = \Gamma[(S_1, \mathcal{P}_r(D)), k_1]$  to **True**.

( $\Leftarrow$ ) To prove other direction, we assume that the algorithm sets  $\Gamma[(S_1, \mathcal{P}_r(A)); k_1]$  to **True**. We argue that  $G[S_1]$  is  $k_1$ -contractible to a grid such that  $\mathcal{P}_r(A)$  is a collection of witness sets in an end-column in a witness structure;  $\Phi(S_1)$  is in  $A$ ; and  $k_1 + |N(S_1)| - r \leq k$ . If  $\Gamma[(S_1, \mathcal{P}_r(A)); k_1]$  is set to **True** in the (*for-loop Initialization*) then, as discussed in first paragraph, this is correct. Consider the case when the value at  $\Gamma[(S_1, \mathcal{P}_r(A)); k_1]$  is set to **True** when the algorithm was processing at index  $[(S_0, \mathcal{P}_r(D)); k_0]$ . Note that value at  $[(S_0, \mathcal{P}_r(D)); k_0]$  has been set **True** by the algorithm as otherwise, it will not change any value while processing this index. Note that  $|A| = a$  and  $|N(S_1)| = b$ . Since  $a$  is a positive integer and  $k_0 + a - r \leq k_1$  (because (*for-loop at Index*) updates only for such values), we know  $|S_0| + k_0 \leq |S_1| + k_1 - 2a + r = z + 1 - 2a + r$ . Since  $a \geq r \geq 1$ , we get  $|S_0| + k_0 \leq z$ . By the induction hypothesis, algorithm has correctly computed value at  $[(S_0, \mathcal{P}_r(D)); k_0]$ . Hence  $G[S_0]$  can be  $k_0$ -contracted to a grid such that  $\Phi(S_0)$  is in  $D$  and there exists a grid witness structure, say  $\mathcal{W}_0$ , such that  $\mathcal{P}_r(D)$  is a collection of witness sets in an end-column. The induction hypothesis also implies and  $k_0 + |N(S_0)| - r \leq k + 3r$ .

Let  $\mathcal{P}_r(A) = \langle A_1, A_2, \dots, A_r \rangle$  be the  $r$ -partition of  $A$  in  $\mathcal{A}_{a,b}[(S_0, \mathcal{P}_r(A))]$  at which *for-loop at Index* changes the value at  $\Gamma[(S_1, \mathcal{P}_r(A)); k_1]$ . By construction, every  $D_i$  in  $\mathcal{P}_r(D)$ ,  $D_i$  is contained in  $A_i$ . Since  $\Phi(S_0)$  is contained in  $D$ , no vertex in  $S_0 \setminus D$  is adjacent with any vertex in  $A$ . Since  $\mathcal{P}_r(A)$  is a  $r$ -slab,  $\mathcal{W}_0 \cup \{A_1, A_2, \dots, A_r\}$  is a grid witness structure of  $G[S_1]$ . Moreover, since  $N(S_0)$  is in  $A$ ,  $\Phi(S_1)$  is contained in  $A$ . Hence,  $G[S_1]$  can be  $k_1$ -contractible to a grid with all vertices in  $\Phi(S)$  in  $A$  and there exists a witness structure for which  $\mathcal{P}_r(A)$  is a collection of witness sets in an end-columns. It remains to argue that  $k_1 + |N(S_1)| - r \leq k$ . We prove this for the case  $k_1 = k_0 + a - r$  as  $k_1 > k_0 + a - r$  case follows from the definition of  $k_1$ -contractibility. Let  $x_0 = |D|$ . As  $x_0$  is the size of an end-column in  $\mathcal{W}_0$ , we have  $x_0 - r \leq k_0$ . As algorithm only considers  $a, b$  such that  $x_0 + a + b \leq k + 3r$ , substituting  $a = k_1 - k_0 + r$  and  $b = |N(S_1)|$  we get  $x_0 + k_1 - k_0 + r + |N(S_1)| \leq k + 3r$ . Using  $x_0 - r \leq k_0$ , we get the desired bound.

This completes the proof of the lemma.  $\blacktriangleleft$

► **Lemma 14.** *Given a graph  $G$  on  $n$  vertices and integers  $k, r$ , the algorithm terminates in time  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$ .*

**Proof.** We first describe an algorithm that given a graph  $G$  on  $n$  vertices and integers  $k, r$ , enumerates all valid tuples in time  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$ . The algorithm computes all  $r$ -slabs in  $G$  which satisfy first property in Definition 10 using Corollary 9. For every  $r$ -slab, it checks whether it satisfies the second property in Definition 10 to determine whether it is a  $k$ -potential  $r$ -slab or not. For a  $k$ -potential  $r$ -slab  $\mathcal{P}_r(D) \equiv \langle D_1, D_2, \dots, D_r \rangle$ , if  $G - D$  has exactly one connected component, say  $C_1$ , then it adds  $(V(C_1) \cup D, \mathcal{P}_r(D))$  and  $(D, \mathcal{P}_r(D))$  to set of valid tuples. If  $G - D$  has two connected components, say  $C_1, C_2$ , then it adds  $(V(C_1) \cup D, \mathcal{P}_r(D))$  and  $(V(C_2) \cup D, \mathcal{P}_r(D))$  to the set of valid tuples. This completes the description of the algorithm. Note that the algorithm returns a set of valid tuples. For a  $k$ -potential  $r$ -slab  $\mathcal{P}_r(D) \equiv \langle D_1, D_2, \dots, D_r \rangle$ ,  $G - D$  has at most two connected components. Hence any  $k$ -potential  $r$ -slab is responsible for at most two nice subsets. By definition of nice subsets, for any nice subset there exists a  $k$ -potential  $r$ -slab responsible for it. Hence the algorithm constructs the set of all valid tuples. The algorithm spends polynomial time for each  $r$ -slab it constructs. Hence, the running time of the algorithm follows from Corollary 9.

The algorithm can compute the table and complete *for-loop Initialization* in time  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$  using the algorithm mentioned in above paragraph. We now argue that the *for-loop Table* takes  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$  time to complete. We partition the set of valid tuples  $\mathcal{V}_k$  using the sizes of the neighborhood of connected component and size of  $r$ -slab in a tuple. For two fixed integers  $x, y$ , define  $\mathcal{V}_k^{x,y} := \{(S, \mathcal{P}_r(D)) \in \mathcal{V}_k \mid |D| \leq x \text{ and } |N(S)| \leq y\}$ . In other words,  $\mathcal{V}_k^{x,y}$  collection of all nice subsets whose neighborhood is of size  $y$  and there is a  $k$ -potential  $r$ -slab of size  $x$  responsible for it. Alternatively,  $\mathcal{V}_k^{x,y}$  is a collection of  $k$ -nice subsets for which there is a  $(x, y)$ - $r$ -slab responsible for it. Since the number of  $(x, y)$ - $r$ -slabs are bounded (Corollary 9) and each  $k$ -potential  $r$ -slab is responsible for at most two nice subsets,  $|\mathcal{V}_k^{x,y}|$  is bounded by  $4^{x+y} \cdot n^{\mathcal{O}(1)}$ .

For each  $(S, \mathcal{P}_r(D)) \in \mathcal{V}_k^{x,y}$ , the algorithm considers every pair of integers  $a(> 0), b(\geq 0)$ , such that  $x + a + b \leq k + 3$  and  $|N(S)| = y \leq a$ , and computes the set  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]$ . By Lemma 8, set  $\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]$  can be computed in time  $4^{a+b-|N(S)|} \cdot n^{\mathcal{O}(1)}$ . The algorithm spends time proportional to  $|\mathcal{A}_{a,b}[(S, \mathcal{P}_r(D))]|$  for *for-loop at Index*. Hence for two fixed integers  $x, y$ , algorithm spends

$$\sum_{\substack{a,b \\ x+a+b \leq k+3r}} 4^{x+y} \cdot 4^{a+b-y} \cdot n^{\mathcal{O}(1)} = \sum_{\substack{a,b \\ x+a+b \leq k+3r}} 4^{x+a+b} \cdot n^{\mathcal{O}(1)} = 4^{k+3r} \cdot n^{\mathcal{O}(1)}$$

time to process all valid tuples in  $\mathcal{V}_k^{x,y}$ . Since there are at most  $\mathcal{O}(k^2)$  feasible values for  $x, y$ , the overall running time of algorithm is bounded by  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$ . This concludes the proof.  $\blacktriangleleft$

The following theorem is implied by Lemmas 13, 14, and the fact that  $(V(G), \mathcal{P}_r(D))$  is a tuple in  $\mathcal{V}_k$  for some  $D$ .

► **Theorem 15.** *There exists an algorithm which given an instance  $(G, k, r)$  of BOUNDED GRID CONTRACTION runs in time  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$  and correctly determines whether it is a YES instance or not. Here,  $n$  is the number of vertices in  $G$ .*

## 5 An FPT algorithm for Grid Contraction

In this section, we present an FPT algorithm for GRID CONTRACTION. Given instance  $(G, k)$  of GRID CONTRACTION is a YES instance if and only if  $(G, k, r)$  is a YES instance of BOUNDED GRID CONTRACTION for some  $r$  in  $\{1, 2, \dots, |V(G)|\}$ . For  $r < 2k + 5$ , we can use algorithm presented in Section 4 to check whether given graph can be contracted to grid



with  $r$  rows or not in FPT time. A choice of this threshold will be clear in the latter part of this section. If algorithm returns YES then we can conclude that  $(G, k)$  is a YES instance of GRID CONTRACTION. If not then we can correctly conclude that if  $G$  is  $k$ -contractible to a grid then the resulting grid has at least  $2k + 5$  rows. This information allows us to find two rows in  $G$  which can safely be contracted. We need the following generalized version of GRID CONTRACTION to state these results formally.

ANNOTATED BOUNDED GRID CONTRACTION

**Parameter:**  $k, r$

**Input:** Graph  $G$ , integers  $k, r, q$ , and a tuple  $(x_1, x_2, x_3, x_4)$  of four different vertices in  $V(G)$

**Question:** Is  $G$   $k$ -contractible to  $\boxplus_{r \times q}$  such that there is a  $\boxplus_{r \times q}$ -witness structure of  $G$  in which the witness sets containing  $x_1, x_2, x_3$ , and  $x_4$  correspond to four corners in  $\boxplus_{r \times q}$ ?

Assume that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  with desired properties via mapping  $\psi$ . Let  $t_1, t_2, t_3$ , and  $t_4$  be corners in  $\boxplus_{r \times q}$  such that  $t_1 \equiv [1, 1], t_2 \equiv [1, q], t_3 \equiv [r, q]$ , and  $t_4 \equiv [r, 1]$ . There are  $4!$  ways in which vertices in  $\{x_1, x_2, x_3, x_4\}$  can be uniquely mapped to corners  $\{t_1, t_2, t_3, t_4\}$ . For the sake of simplicity, we assume that we are only interest in the case in which  $x_1, x_2, x_3, x_4$  are mapped to  $t_1, t_2, t_3$ , and  $t_4$  respectively. In other words,  $\psi(x_i) = t_i$  for all  $i \in \{1, 2, 3, 4\}$ .

We can modify the algorithm presented in Section 4 obtain an algorithm for ANNOTATED BOUNDED GRID CONTRACTION problem which is fixed parameter tractable when parameterized by  $(k + r)$ . The modified algorithm only initializes tuple  $(S, \mathcal{P}_r(D)) \in \mathcal{V}_k$  such that  $S = D$ ,  $k' \geq |S| - r = |D| - r$ , and  $x_1, x_2$  are in first and last parts in  $\mathcal{P}_r(D)$  in the (*for-loop Initialization*) step. Recall that the algorithm in Section 4 set  $\Gamma[(S, \mathcal{P}_r(D)); k'] = \text{True}$  if and only if  $k' + |N(S)| - r \leq k$  and there is a  $(r \times q')$ -grid witness structure of  $G[S]$ , for some integer  $q'$ , such that  $\mathcal{P}_r(D)$  is collection of witness sets in an end-column and  $\Phi(S)$  is in  $D$ . Instead of storing **True** or **False**, the modified algorithm stores  $q'$  if it is **True** and 0 otherwise. With these simple modifications, we obtain the following result.

► **Lemma 16.** *There exists an algorithm which given an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  of ANNOTATED BOUNDED GRID CONTRACTION runs in time  $4^{k+3r} \cdot n^{\mathcal{O}(1)}$  and correctly determines whether it is a YES instance or not. Here,  $n$  is the number of vertices in  $G$ .*

In the case, when  $r < 2k + 5$  the algorithm mentioned in the above lemma is fixed parameter tractable when the parameter is  $k$  alone. When  $r \geq 2k + 5$ , we argue that if  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  is a YES instance then there exists a *horizontal decomposition* of  $G$  (Lemma 18). We formally define horizontal decomposition as follows.

► **Definition 17** (Horizontally-Decomposable). *Consider an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  of ANNOTATED BOUNDED GRID CONTRACTION. A graph  $G$  is said to be horizontally-decomposable if  $V(G)$  can be partitioned into four non-empty parts  $C_{12}, S_u, S_v$ , and  $C_{34}$  which satisfies following properties.*

- The graphs  $G[C_{12}], G[C_{34}]$  are connected and  $x_1, x_2 \in C_{12}, x_3, x_4 \in C_{34}$ .
- The graph  $G[S_u \cup S_v]$  is a  $2 \times q$  grid with  $S_u, S_v$  correspond to vertices in its two rows.
- $C_{12}$  and  $C_{34}$  are the two connected components of  $G \setminus (S_u \cup S_v)$ .
- $N(C_{12}) = S_u$  and  $N(C_{34}) = S_v$ .

► **Lemma 18.** *Consider an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  of ANNOTATED BOUNDED GRID CONTRACTION such that  $2k + 5 \leq r$ . If it is a YES instance then there exists a horizontal decomposition of  $G$ .*

**Proof.** Assume that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  with desired properties via mapping  $\psi$ . By Observation 2.2, there are at most  $k$  big-witness sets. This implies that there are at most  $k$  rows in  $\boxplus_{r \times q}$  which contain vertices corresponding to big-witness sets. Since there are at least  $2k + 5$  rows in  $\boxplus_{r \times q}$ , there exists  $i_o$  in  $\{2, 3, \dots, r - 2\}$  such that no vertex in  $i_o^{th}$  and  $(i_o + 1)^{th}$  row corresponds to a big witness set. Define  $C_{12}, S_u, S_v, C_{34}$  as follows.

- $C_{12} := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i < i_o \text{ and } j \in [q]\}$
- $S_u := \{x \in V(G) \mid \psi(x) = [i_o, j] \text{ for some } j \in [q]\}$ .
- $S_v := \{x \in V(G) \mid \psi(x) = [i_o + 1, j] \text{ for some } j \in [q]\}$
- $C_{34} := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i > i_o + 1 \text{ and } j \in [q]\}$

It is easy to verify that  $(C_{12}, S_u, S_v, C_{34})$  is a horizontal decomposition of  $G$ . ◀

Consider an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$ , let  $(C_{12}, S_u, S_v, C_{34})$  be a horizontal decomposition of  $G$ . Reduction Rule 5.1 contracts all the edges across  $S_u, S_v$ . Note that in the resulting instance,  $r$  is decreased by one.

▷ **Reduction Rule 5.1.** For an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$ , let  $(C_{12}, S_u, S_v, C_{34})$  be a horizontal decomposition of  $G$ . Let  $S_u = \{u_1, u_2, \dots, u_q\}$  and  $S_v = \{v_1, v_2, \dots, v_q\}$ . Let  $G'$  be the graph obtained from  $G$  by contracting all the edges in  $\{u_j v_j \mid j \in [q]\}$ . Return instance  $(G', k, r - 1, q, (x_1, x_2, x_3, x_4))$ .

As  $S_u, S_v$  are  $\{(x_1 - x_4), (x_2 - x_3)\}$ -separators in  $G$ , by Observation 2.2, sets  $\psi(S_u), \psi(S_v)$  are  $\{(t_1 - t_4), (t_2 - t_3)\}$ -separators in  $\boxplus_{r \times q}$ . We argue that  $\psi(S_u)$  and  $\psi(S_v)$  correspond to two consecutive rows and it was safe to contract edges across  $S_u, S_v$ .

► **Lemma 19.** *Reduction Rule 5.1 is safe.*

**Proof.** Note that  $\boxplus_{(r-1) \times q}$  can be obtained from  $\boxplus_{r \times q}$  by contracting all edges across any two consecutive rows. Also, this operation does not remove any vertex from witness sets corresponding to corner vertices in grid.

( $\Rightarrow$ ) Assume  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  via mapping  $\psi$  with desired properties. We argue that  $G'$  is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$  with desired properties. Let  $t_1, t_2, t_3$ , and  $t_4$  be the four corners of  $\boxplus_{r \times q}$  such that  $\psi(x_i) = t_i$  for all  $i \in \{1, 2, 3, 4\}$ . By Observation 2.2,  $\psi(S_u), \psi(S_v)$  are  $\{(t_1, t_4), (t_2, t_3)\}$ -separators in  $\boxplus_{r \times q}$ . By Observation 2.1,  $|\psi(S_u)|, |\psi(S_v)| \geq q$ . By the property of mapping  $\psi$ , we have  $|\psi(S_u)| \leq |S_u|$  and  $|\psi(S_v)| \leq |S_v|$ . Since  $|S_u| = |S_v| = q$ , we have  $|\psi(S_u)| = |\psi(S_v)| = q$ . Hence, by Observation 2.1,  $\psi(S_u)$  and  $\psi(S_v)$  corresponding to rows in  $\boxplus_{r \times q}$ . Let  $\psi(S_u)$  and  $\psi(S_v)$  correspond to rows  $i_1, i_2$ . Since there are multiple edges across  $S_u, S_v$ , we have  $|i_1 - i_2| \leq 1$ . As  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$ , if  $|i_1 - i_2| = 1$  then  $G'$  is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$ . If  $|i_1 - i_2| = 0$  then  $G'$  is  $(k - q)$ -contractible to  $\boxplus_{r \times q}$  as  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  and  $G' = G / \{v_j u_j \mid j \in [q]\}$ . Hence,  $G'$  is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$ .

( $\Leftarrow$ ) Let  $S_o = \{s_1, s_2, \dots, s_q\}$  be the set vertices in  $G'$  which are obtained by contracting edges  $u_j v_j$  in  $G$ . In other words, for  $j$  in  $[q]$ , let  $s_j$  be the new vertex added while contracting edge  $u_j v_j$ . Note that  $S_o$  is  $\{(x_1 - x_4), (x_2 - x_3)\}$ -separator in  $G'$ .

Assume that  $G'$  is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$  with the desired properties via mapping  $\phi$ . Let  $t_1, t_2, t_3$ , and  $t_4$  be the four corners of  $\boxplus_{(r-1) \times q}$  such that  $\phi(x_i) = t_i$  for  $i \in \{1, 2, 3, 4\}$ . By Observation 2.2,  $\phi(S_o)$  is a  $\{(t_1, t_4), (t_2, t_3)\}$ -separators in  $\boxplus_{(r-1) \times q}$ . By Observation 2.1,  $|\phi(S_o)| \geq q$ . By the property of mapping  $\phi$ , we have  $|\phi(S_o)| \leq |S_o|$ . Since  $|S_o| = q$ , we have  $|\phi(S_o)| = q$ . Hence, by Observation 2.1,  $\phi(S_o)$  corresponding to a row, say  $r_o$ , in  $\boxplus_{(r-1) \times q}$ .

As  $G$  is  $q$ -contractible to  $G'$  (which is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$ ), we know that  $G$  is  $(k + q)$ -contractible to  $\boxplus_{(r-1) \times q}$ . A mapping  $\psi : V(G) \rightarrow \boxplus_{r \times q}$  as follows corresponds to this contraction. For every  $x$  in  $V(G) \setminus (S_u \cup S_v)$ , define  $\psi(x) = \phi(x)$  and for every  $x$  in  $\{u_j, v_j\}$ , define  $\psi(x) = \phi(s_j)$ . We argue that  $r_o^{th}$  row in  $\boxplus_{(r-1) \times q}$  is partible. For every  $j$  in

$[q]$ , we define a partition  $U_j, V_j$  of  $\psi^{-1}([i_1, j])$  which satisfy all the properties mentioned in Definition 2.

For  $j$  in  $[q]$ , let  $X_j = \psi^{-1}([i_o, j])$ . As  $s_j$  was present in  $\phi^{-1}([i_o, j])$ , vertices  $u_j, v_j$  are present in  $X_j$ . By the property of  $\phi$ , set  $\phi^{-1}([i_o, j])$  is connected in  $G'$ . Since vertex  $s_j$  is obtained from contracting edge  $u_j v_j$  in  $G$ , graph  $G[X_j]$  is connected. Since  $\phi(S_o)$  corresponds to a row of with  $q$  vertices and  $|S_o| = q$ , vertices  $u_{j'}, v_{j'}$  are present in  $X_j$  if and only if  $j' = j$ . In other words,  $X_j \cap (S_u \cup S_v) = \{u_j, v_j\}$ . Define  $U_j := (X_j \cap C_{12}) \cup \{u_j\}$  and  $V_j := (X_j \cap C_{34}) \cup \{v_j\}$ . Since  $(C_{12}, S_u, S_v, C_{34})$  is a horizontal decomposition of  $G$ , sets  $U_j, V_j$  is a non-empty partition of  $X_j$ . We argue that  $U_j, V_j$  satisfy all the properties in Definition 2. As  $N(C_{12}) = S_u$ , no vertex in  $U_j \setminus \{u_j\}$  is adjacent with  $S_v \cup C_{34}$ . By similar arguments, no vertex in  $V_j \setminus \{v_j\}$  is adjacent with  $S_u \cup C_{12}$ . As  $G[X_j]$  is connected and  $U_j \subseteq S_u \cup C_{12}; V_j \subseteq S_v \cup C_{34}$ , graphs  $G[U_j], G[V_j]$  are connected. Moreover, for  $j' \in [q]$ , sets  $U_j, V_{j'}$  are adjacent if and only if  $u_j, v_{j'}$  are adjacent. Since  $G[S_1 \cup S_2]$  is a  $(2 \times q)$ -grid,  $u_j$  and  $v_{j'}$  are adjacent if and only if  $j = j'$ . Hence  $U_j$  and  $V_{j'}$  are adjacent if and only if  $j = j'$ . Since  $U_j \subseteq X_j$  and  $U_{j'} \subseteq X_{j'}$ ,  $U_j$  and  $U_{j'}$  are non adjacent if  $|j - j'| > 1$ . If  $|j - j'| = 1$  then  $U_j, U_{j'}$  are adjacent as they contain  $u_j$  and  $u_{j'}$ . Hence  $U_j, U_{j'}$  are adjacent if and only if  $|j - j'| = 1$ . By similar arguments,  $V_j, V_{j'}$  are adjacent if and only if  $|j - j'| = 1$ . As no vertex in  $U_j$  is adjacent with  $C_{34}$  and no vertex in  $V_j$  is adjacent with  $C_{12}$ , we can conclude that partition  $U_j, V_j$  satisfy all the properties in Definition 2.

Since  $r^{th}$  row in  $\boxplus_{(r-1) \times q}$  is partible, Lemma 3 implies that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$ . This concludes the proof of reverse direction.

Hence  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  is a YES instance of ANNOTATED BOUNDED GRID CONTRACTION if and only if  $(G', k, r - 1, q, (x_1, x_2, x_3, x_4))$  is a YES instance. ◀

It remains to argue that Reduction Rule 5.1 can be implemented in polynomial time. In Lemma 21, we argue there exists an algorithm that can find a horizontal decomposition, if exists, in polynomial time. We use the following structural lemma to prove the previous statement.

► **Lemma 20.** *Given two adjacent vertices  $u_1, v_1$  in  $G$ , there is at most one subset  $S$  of  $V(G)$  such that (a)  $G[S]$  is a  $(2 \times q)$  grid, (b)  $u_1, v_1$  are two vertices in the first column of  $G[S]$ , and (c) each row in  $S$  is a separator in  $G$ . Moreover, if such a subset exists then it can be found in polynomial time.*

**Proof.** For the sake of a contradiction, assume that there are two such subsets, say  $S, S'$ , of  $V(G)$ . Let  $S_u, S_v$  and  $S'_u, S'_v$  be the two rows in  $G[S]$  and  $G[S']$ , respectively. Let  $j$  be the first column in which vertices in  $S, S'$  differs. As the first columns in  $S, S'$  are same,  $j \geq 2$ . Let  $u_j, v_j$  and  $u'_j, v'_j$  be the vertices in  $j^{th}$  row of  $S$  and  $S'$  respectively. Without loss of generality, assume  $u_j \neq u'_j$ . Since  $u_j, u'_j$  both are adjacent with  $u_{j-1}$  and  $u_j \in S_u$ , we can conclude  $u'_j \notin S_u$ . The only vertex in  $S_v$  which is adjacent with  $u_{j-1}$  is  $v_{j-1}$  and  $u'_j \neq v_{j-1}$ , we have  $u'_j \notin S_v$ . By similar argument, we can prove that  $v'_j \notin S_u$ . As  $v_{j-1}$  is in  $S_v$ , we have  $v_{j-1} \notin S_u$ . To summarize, we can conclude that  $u'_j \notin S_v$  and  $u'_j, v_{j-1}, v'_j \notin S_u$ .

Consider separators  $S_u, S_v$  in graph  $G$ . Let  $C$  be a connected component of  $G - S_u$  which contains  $S_v$ . Since  $S_v$  is also a separator, the only vertices in  $C$  which are adjacent with  $S_u$  are in  $S_v$ . This implies that vertex  $u'_j$  which is adjacent with  $S_u$  can not be in  $C \setminus S_v$ . Since  $u'_j \notin S_v$ , we can conclude that  $u'_j$  and  $S_v$  are in different connected components of  $G - S_u$ . This implies  $u'_j$  and  $v_{j-1}$  are in different connected component of  $G - S_u$ . But,  $u'_j, v_{j-1}, v'_j \notin S_u$  and there exists a path  $(u'_j, v'_j, v_{j-1})$  in  $G$ . This leads to a contradiction to the fact that  $u'_j$  and  $v_{j-1}$  are in different connected components of  $G - S_u$ . Hence our assumption is wrong and there exists at most one such set.

Given  $u_1, v_1$  and the uniqueness of a subset with the desired property, if there exists such subgraph then there are unique choices for vertices in the second column. In other words, a subset  $S$  with desired properties exists if and only if there is a unique pair of adjacent vertices, say  $u_2, v_2$ , in graph  $G$  which satisfy following conditions – (1)  $u_1u_2, v_1v_2 \in E(G)$  and (2)  $u_1v_2, v_1u_2 \notin E(G)$ . One can stepwise add new columns in  $S$  while checking  $G[S]$  remains a grid with two rows until no more columns can be added. This algorithm terminates in polynomial time and either returns a subset with desired properties or correctly concludes that no such subgraph exists. ◀

► **Lemma 21.** *There exists an algorithm which given an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  of ANNOTATED BOUNDED GRID CONTRACTION runs in polynomial time and either returns a horizontal decomposition of  $G$  or correctly concludes that no such decomposition exists.*

**Proof.** For every pair of adjacent vertices  $u_1, v_1$  in  $G$ , the algorithm tries to find a subset of  $V(G)$  with the properties mentioned in the statement of Lemma 20. If such a subset exists, say  $S$ , then the algorithm checks whether  $S$  and connected components of  $G - S$  satisfy the conditions mentioned Definition 17. The algorithm returns a horizontal decomposition if it finds one. As the algorithm exhaustively searches for all possible  $(2 \times q)$ -grids which are also separators, if it does not return a horizontal decomposition then the graph does not admit a horizontal decomposition. The running time of the algorithm is implied by the fact that algorithm runs over all edges in the input graph, conditions in Definition 17 can be checked in polynomial time, and by Lemma 20. ◀

We are now in a position to present main result of this section.

► **Theorem 22.** *There exists an algorithm which given an instance  $(G, k)$  of GRID CONTRACTION runs in time  $4^{6k} \cdot n^{\mathcal{O}(1)}$  and correctly determines whether it is a YES instance or not. Here,  $n$  is the number of vertices in  $G$ .*

**Proof.** The algorithm starts with checking whether graph  $G$  is  $k$ -contractible to a path using the algorithm in [15]. If it is then the algorithm returns YES else it creates polynomially many instances of ANNOTATED BOUNDED GRID CONTRACTION by guessing all possible values of  $r, q, x_1, x_2, x_3, x_4$ . It processes these instances with increasing values of  $r$ . Ties are broken arbitrarily. For  $r < 2k + 5$ , the algorithm check whether  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  is a YES instance of ANNOTATED BOUNDED GRID CONTRACTION using Lemma 16. For  $r \geq 2k + 5$ , the algorithm checks whether there exists a horizontal decomposition of  $G$  using Lemma 21. If there exists a horizontal decomposition of  $G$  then the algorithm applies Reduction Rule 5.1 to obtain another instance of ANNOTATED BOUNDED GRID CONTRACTION with a smaller value of  $r$ . The algorithm repeats the above step until  $r < 2k + 5$  or the graph in a reduced instance does not have a horizontal decomposition. In the first case, it checks whether a reduced instance is a YES instance or not using Lemma 16. In the second case, it continues to the next instance created at the start of the algorithm. The algorithm returns YES if at least one of the instances of ANNOTATED BOUNDED GRID CONTRACTION is a YES instance.

It is easy to see that an instance  $(G, k)$  of GRID CONTRACTION is a YES instance if and only if there exists integers  $r, q$  in  $\{1, 2, \dots, |V(G)|\}$  and four vertices  $x_1, x_2, x_3, x_4$  in  $V(G)$  such that  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  is a YES instance of ANNOTATED BOUNDED GRID CONTRACTION. Lemma 19 implies the correctness of the step where the algorithm repeatedly applies Reduction Rule 5.1 and check whether the reduced instance is a YES instance of ANNOTATED BOUNDED GRID CONTRACTION or not. Consider an instance  $(G, k, r, q, (x_1, x_2, x_3, x_4))$  such that  $r > 2k + 5$  and there is no horizontal decomposition of  $G$ . By Lemma 18, the algorithm correctly concludes that it is a NO instance and continues

to the next instance. This implies the correctness of the algorithm. The running time of the algorithm is implied by Lemmas 16, 21 and the fact that the algorithm presented in [15] runs in time  $2^{k+o(k)} \cdot n^{\mathcal{O}(1)}$ .  $\blacktriangleleft$

## 6 NP-Completeness and Lower Bounds

In this section, we prove that GRID CONTRACTION problem is NP-Complete. We also argue that the dependency on the parameter in the running time of the algorithm presented in Section 5 is optimal, up to constant factors in the exponent, under a widely believed hypothesis. We define the problems mentioned in this paragraph in the latter parts. Brouwer and Veldman presented a reduction from HYPERGRAPH 2-COLORABILITY problem to  $H$ -CONTRACTION problem [5]. We present a reduction from NAE-SAT problem to HYPERGRAPH 2-COLORABILITY problem. We argue that the reduction used by Brouwer and Veldman can be used to reduce the HYPERGRAPH 2-COLORABILITY problem to GRID CONTRACTION problem. Using these reductions and the fact there is no *sub-exponential* time algorithm for NAE-SAT, we obtain desired results.

We start with the definition of HYPERGRAPH 2-COLORABILITY problem. An hyper-edge is called monochromatic if all vertices in this edge has the same color. In HYPERGRAPH 2-COLORABILITY problem, an input is a hypergraph  $\mathcal{H}$  and the objective is to partition  $V(\mathcal{H})$  into two colors such that every edge in  $E(\mathcal{H})$  is monochromatic. For a fixed graph  $H$ , the  $H$ -CONTRACTION problem takes a graph  $G$  as an input and the objective is to determine whether  $G$  can be contracted to  $H$  or not. Brouwer and Veldman proved the following result.

► **Proposition 23** (Theorem 9 [5]). *If  $H$  is a connected triangle free graph other than a star then  $H$ -CONTRACTION is NP-Complete.*

We are interested in the case when  $H$  is a cycle on four vertices (which is denoted by  $C_4$ ). We present the reduction that is used to prove the above proposition in [5]. For the sake of simplicity, we restrict the reduction to the case when  $H = C_4$ . Without loss of generality, we can assume that any instance  $\mathcal{H}$  of HYPERGRAPH 2-COLORABILITY contains at least two edges and has a hyper-edge which contains all vertices in  $\mathcal{H}$ .

**Reduction-(1):** Given a hypergraph  $\mathcal{H}$  the reduction produces a graph  $G$ , an instance of  $C_4$ -CONTRACTION, as follows:

- For every vertex  $x$  in  $V(\mathcal{H})$ , it adds vertex  $v_x$ . Let  $X$  be the set of all vertices corresponding to vertices in the hypergraph.
- For every hyper-edge  $e$  in  $E(\mathcal{H})$ , it adds two vertices  $e_1$  and  $e_2$ . Let  $E_1, E_2$  be the collection of  $e_1$ s and  $e_2$ s for all edges in the hypergraph.
- It adds two special vertices  $v_1$  and  $v_2$ .
- It adds all edges between every pair of vertices in  $X$ . In other words, the algorithm converts  $X$  into a clique.
- It adds all edges between every pair of vertices  $(e_1, e'_2)$ . In other words, the algorithm converts  $G[E_1 \cup E_2]$  into a complete bipartite graph with  $E_1, E_2$  as its two maximal independent sets.
- For a vertex  $v_x$  in  $X$  and a vertex  $e_i$  in  $E_i$ , if edge  $e$  contains vertex  $x$  then the algorithm adds edge  $v_x e_i$ . Here,  $i \in \{1, 2\}$ .
- It adds edges to make  $v_1$  adjacent with every vertex in  $E_1$  and  $v_2$  adjacent with every vertex in  $E_2$ .
- It adds edge  $v_1 v_2$ .

In the following claim, we argue that the above reduction can be used to prove a reduction from HYPERGRAPH 2-COLORABILITY to GRID CONTRACTION.

▷ **Claim 24.** Let  $G$  be the graph returned by Reduction (1). Graph  $G$  is a YES instance of  $C_4$ -CONTRACTION if and only if  $(G, |V(G)| - 4)$  is a YES instance of GRID CONTRACTION.

**Proof.** As  $C_4$  is a  $(2 \times 2)$ -grid, the forward direction of the lemma is true. Observe that the diameter of  $G$  is two. Let  $k = |V(G)| - 4$ . As an edge contraction reduces the number of vertices by exactly one, if  $G$  is  $k$ -contractible to a grid then the resulting grid has at least four vertices. Moreover, for any graph  $G'$  and an edge  $e$  in it, the diameter of  $G'/e$  is at most the diameter of  $G$ . Hence,  $G$  is  $k$ -contractible to a grid that has at least four vertices and has a diameter two. Only a  $(2 \times 2)$ -grid satisfy both of these properties. This proves the forward direction and completes the proof of the claim. ◀

Using Proposition 23 and Claim 24, we obtain the following result.

► **Lemma 25.** Assume that Reduction (1) constructs graph  $G$  when an input is hypergraph  $\mathcal{H}$ . Then,  $\mathcal{H}$  is a YES instance of HYPERGRAPH 2-COLORABILITY if and only if  $(G, |V(G)| - 4)$  is a YES instance of GRID CONTRACTION.

In the remaining section, we present a reduction from NAE-SAT to HYPERGRAPH 2-COLORABILITY. In SAT, we are given a conjunctive normal formula and the objective is to find an assignment that evaluates the formula to **True**. 3-SAT is a restricted version of SAT in which every clause contains at most three variables. In NAE-SAT variation of the problem, the objective is to find a satisfying assignment of variables such that for any clause all of its variables are not set to **True**. A simple reduction from 3-SAT to NAE-SAT is as follows: Given an instance  $\phi$  of 3-SAT, add a new variable, say  $x$ , and replace every clause  $C$  in  $\phi$  by  $C \wedge x$ . Add a clause  $(\bar{x})$  to  $\phi$  to obtain an instance  $\phi'$  of NAE-SAT. It is easy to verify that  $\phi$  is a YES instance of 3-SAT if and only if  $\phi'$  is a YES instance of NAE-SAT. Moreover, the summation of the number of variables and the number of clauses in  $\phi'$  is two more than the sum of the number of variables and the number of clauses in NAE-SAT. Let  $N, M$  be the number of variables and the number of a clause, respectively. It is known that unless Exponential Time Hypothesis (ETH) fails 3-SAT problem can not be solved in time  $2^{o(N+M)}$  [16]. The above reduction implies that unless ETH fails NAE-SAT can not be solved in time  $2^{o(N+M)}$ . We now present a reduction from NAE-SAT to HYPERGRAPH 2-COLORABILITY.

**Reduction-(2):** Given an instance  $\phi$  of NAE-SAT, the reduction algorithm constructs a hypergraph, say  $\mathcal{H}$ , as follows: For every variable  $x$ , add two vertices  $x_{pos}, x_{neg}$ . For every variable  $x$ , add a hyper-edge  $\{x_{pos}, x_{neg}\}$ . For every clause, add a hyper-edge between the literals present in the clause. For example, for a clause  $x \wedge \bar{y} \wedge \bar{z} \wedge w$  add hyper-edge  $\{x_{pos}, y_{neg}, z_{neg}, w_{pos}\}$ .

► **Lemma 26.** Assume that Reduction (2) constructs hypergraph  $\mathcal{H}$  when an input is formula  $\phi$ . Then,  $\phi$  is a YES instance of NAE-SAT if and only if  $\mathcal{H}$  is a YES instance of HYPERGRAPH 2-COLORABILITY.

**Proof.** In forward direction, let  $\psi$  be a satisfying assignment of variables in  $\phi$  such that for any clause in  $\phi$ , not all the literals are set to **True**. We construct a coloring function  $\lambda : V(\mathcal{H}) \rightarrow \{0, 1\}$  as follows: For a variable  $x$ , if  $\psi$  assigns  $x$  to **True** then  $\lambda(x_{pos}) = 1$  and  $\lambda(x_{neg}) = 0$ . If  $\psi$  assigns  $x$  to **False** then  $\lambda(x_{pos}) = 0$  and  $\lambda(x_{neg}) = 1$ . Every edge of the



type  $\{x_{pos}, x_{neg}\}$ , contains a vertex which is colored 0 and 1. For every edge corresponding to a clause has a vertex which is colored 1 (as  $\psi$  is a satisfying assignment) and a vertex colored 0 (as  $\psi$  does not set all literals to **True**). This implies that  $\mathcal{H}$  is a YES instance of HYPERGRAPH 2-COLORABILITY.

In reverse direction, let  $\lambda : V(\mathcal{H}) \rightarrow \{0, 1\}$  be a 2-coloring of  $V(\mathcal{H})$  such that every edge contains vertices with both colors. We construct an assignment  $\psi$  for formula  $\phi$ . For a vertex  $x_{pos}$ , if  $\lambda(x_{pos}) = 1$  then  $\psi$  assigns  $x$  to **True**. If  $\lambda(x_{pos}) = 0$  then  $\psi$  assigns  $x$  to **False**. We first argue that  $\psi$  is a proper assignment for variables in  $\phi$ . Consider a hyper-edge  $\{x_{pos}, x_{neg}\}$ . Since every edge has vertices with both colors, if  $\lambda(x_{pos}) = 1$  then  $\lambda(x_{neg}) = 0$ . This implies if  $\psi$  assign  $x$  to **True** at some point, then it never assigns it to **False**. Consider a hyper-edge corresponding to a clause. Since there is a vertex with color 1 in this edge,  $\phi$  assigned **True** to at least one literal appearing in the clause. Similarly, since there is a vertex with color 0 in this edge,  $\phi$  assigned **False** to at least one literal appearing in the clause. Hence,  $\psi$  is a satisfying assignment for  $\phi$  and there is no clause in  $\phi$  for which  $\psi$  assigns all literals to **True**. This implies that  $\phi$  is a YES instance of NAE-SAT. ◀

Observe that given an instance  $\phi$  of NAE-SAT with  $N$  variable and  $M$  clauses, Reduction (2) constructs a graph  $\mathcal{H}$  with  $2N$  vertices and  $M + N$  edges. This implies unless ETH fails, HYPERGRAPH 2-COLORABILITY can not be solved in time  $2^{o(n'+m')}$ , where  $n', m'$  are the number of vertices and the number of edges in an input graph. Note that given a hyper-graph  $\mathcal{H}$  on  $n'$  vertices and  $m'$  hyper-edges, the Reduction (1) constructs a graph on  $n' + 2m' + 2$  vertices. This leads to the main result of this section.

► **Theorem 27.** GRID CONTRACTION is NP-Complete. Moreover, unless ETH fails, it can not be solved in time  $2^{o(n)}$ , where  $n$  is the number of vertices in an input graph.

## 7 Kernelization

In this section, we present a polynomial kernel for the GRID CONTRACTION problem. In Section 5, we reduced an instance of GRID CONTRACTION to polynomially many instances of ANNOTED BOUNDED GRID CONTRACTION such that the original instance is a YES instance if and only one of these instances is a YES instance. One can argue that exhaustively application of Reduction Rule 5.1 leads to a *Turing Compression*<sup>1</sup> of the size  $\mathcal{O}(k^2)$ . We use a similar approach, but with weaker bounds, to obtain a kernel of size  $\mathcal{O}(k^4)$ .

If the input graph is not connected then we can safely conclude that we are working with a NO instance. The following reduction rule checks two more criteria in which it is safe to return a NO instance.

▷ **Reduction Rule 7.1.** For an instance  $(G, k)$ , if

- there exists a vertex in  $G$  whose degree is more than  $k + 5$ , or
- there are  $6k + 1$  vertices in  $G$  whose degrees are more than 5,

then return a trivial NO instance.

► **Lemma 28.** Reduction Rule 7.1 is safe.

**Proof.** The maximum degree of a vertex in a grid is four. An edge contraction can reduce the number of vertices by one. If a vertex has a degree more than  $k + 5$  in  $G$  then even after

<sup>1</sup> Please see, for example, [10, Chapter 22] for formal definition.

## 23:22 On the Parameterized Complexity Of Grid Contraction

$k$  edge contractions, its degree is at least five. Hence, in this case,  $G$  is not  $k$ -contractible to a grid.

Every vertex of degree five or more in  $G$  is either in a big-witness set or it is a singleton witness set which is adjacent with a big witness set. By Observation 2.2, there are at most  $k$  big witness sets which contains at most  $2k$  vertices. The big witness sets can be adjacent with at most  $4k$  singleton witness set. Hence  $G$  can have at most  $6k$  vertices which has a degree more than five.  $\blacktriangleleft$

We define  $k_o = (4k + 8) \cdot (k + 1) + 1$ . Consider an instance  $(G, k)$  on which Reduction Rule 7.1 is not applicable. If  $G$  has at most  $k_o^2 + k + 1$  vertices then we can argue that we have a kernel of the desired size. Consider a case when  $|V(G)| \geq k_o^2 + k + 1$ . We argue that in this case, if  $(G, k)$  is a YES instance then there exists a large grid separator in a graph  $G$  (Lemma 30).

► **Definition 29** ( $(p \times t)$ -grid-separator). Consider an instance  $(G, k)$  of GRID CONTRACTION. A subset  $S$  of  $V(G)$  is called a  $(p \times t)$ -grid-separator of  $G$  if it has following three properties.

- $G[S] = \Gamma_{p \times t}$ .
- Graph  $G - S$  has exactly two connected components, say  $C_1$  and  $C_2$ .
- $|V(C_1)|, |V(C_2)| \geq k + 1$  and  $N(C_1) = R_1, N(C_2) = R_p$ , where  $R_1, R_p$  are the first and last row in  $G[S]$ .

► **Lemma 30.** Consider an instance  $(G, k)$  of GRID CONTRACTION such that  $|V(G)| \geq k_o^2 + k + 1$ . If  $(G, k)$  is a YES instance then there exists a  $((4k + 6) \times t)$ -grid-separator in  $G$  for some integer  $t$ .

**Proof.** Assume that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$  via mapping  $\psi$ . Without loss of generality, we can assume that  $r \geq q$ . Since any edge contraction can reduce number of vertices in  $G$  by one, the number of vertices in  $\boxplus_{r \times q}$  is at least  $k_o^2 + 1$ . This implies  $r \cdot q \geq k_o^2 + 1$ . Since  $r \geq q$ , we have  $r \geq k_o$ . We show that there exists a partition of  $V(G)$  into  $C_1, S, C_2$  such that these sets satisfy properties mentioned in Definition 29.

To satisfy the cardinality condition, we include all vertices in first  $(k + 1)$  many rows in  $C_1$  and last  $(k + 1)$  many rows in  $C_2$ . Note that there are still at least  $k_o - 2(k + 1) = (4k + 6) \cdot (k + 1) + 1$  rows in the middle. By Observation 2.2, there are at most  $k$  big-witness sets. This implies that there are at most  $k$  rows in  $\boxplus_{r \times q}$  which contain vertices corresponding to big-witness sets. Hence, there are at least  $(4k + 5) \cdot (k + 1) + 1$  rows which does not contain any big witness set. The rows with big witness set partition the rows without any big witness set into at most  $(k + 1)$  parts such that each part is connected. At least one of these parts must have  $(4k + 6)$  rows. Hence, there exists  $i_o$  in  $\{k + 2, k + 3, \dots, r - (k + 2)\}$  such that no vertex in  $i_o^{th}$  to  $(i_o + 4k + 6)^{th}$  rows corresponds to a big witness set. Define  $C_1, S, C_2$  as follows.

- $C_1 := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i < i_o \text{ and } j \in [q]\}$
- $S := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i \in \{i_o, i_o + 1, \dots, i_o + 4k + 6\} \text{ and } j \in [q]\}$ .
- $C_2 := \{x \in V(G) \mid \psi(x) = [i, j] \text{ for some } i > i_o + 4k + 6 \text{ and } j \in [q]\}$

It is easy to verify that  $C_1, S, C_2$  satisfy all the properties mentioned in Definition 29.  $\blacktriangleleft$

In the following lemma, we argue that the existence of such a large grid separator in a graph implies certain restrictions on the grid to which the graph can be contracted.

► **Lemma 31.** Consider an instance  $(G, k)$  of GRID CONTRACTION. Let  $S$  be a  $((4k + 5) \times t)$ -grid-separator of  $G$ . If  $G$  is  $k$ -contractible to a grid  $\boxplus_{r \times q}$  then  $q = t$ .

**Proof.** Assume that  $G$  is  $k$ -contractible to  $\boxtimes_{r \times q}$  via mapping  $\psi$ . Note that this implies  $G$  is  $k$ -contractible to  $\boxtimes_{q \times r}$  via mapping  $\psi'$ . Rows (and corresponding witness sets) in  $\boxtimes_{r \times q}$  correspond to columns (and corresponding witness sets) in  $\boxtimes_{q \times r}$  and vice-versa.

By Observation 2.2, there are at most  $k$  big-witness sets. This implies that there are at most  $2k$  rows in  $G[S] \equiv \boxtimes_{(4k+5) \times t}$  which contain vertices which are part of big-witness sets. Since there are  $(4k+5)$  rows, there exists  $i_o$  in  $\{2, 3, \dots, 4k+5-2\}$  such that no vertex in  $i_o^{th}$  and  $(i_o+1)^{th}$  row corresponds to a big witness set. Let  $R_{i_o} (= \{u_1, u_2, \dots, u_t\})$  and  $R_{i_o+1} (= \{v_1, v_2, \dots, v_t\})$  be the  $i_o^{th}$  and  $(i_o+1)^{th}$  rows in  $G[S]$ . Since no edge incident on vertices in  $R_{i_o} \cup R_{i_o+1}$  is begin contracted, we can conclude following two things: (a) For  $j, j'$  in  $[t]$ , vertices  $\psi(u_j)$  and  $\psi(u_{j'})$  (similarly,  $\psi(v_j)$  and  $\psi(v_{j'})$ ) are adjacent with each other if and only if  $|j - j'| = 1$ . (b) For  $j, j'$  in  $[t]$ , vertices  $\psi(u_j)$  and  $\psi(v_{j'})$  are adjacent with each other if and only if  $j = j'$ . This implies that  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$  are the vertices contained in two consecutive rows or columns in  $\boxtimes_{r \times q}$ . If  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$  are in two consecutive columns then we repeats the arguments with mapping  $\psi'$ . Because of symmetry, we can assume that vertices in  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$  are in two consecutive rows in  $\boxtimes_{r \times q}$ . Let  $i'$  and  $i' + 1$  be the rows in  $\boxtimes_{r \times q}$  which contains vertices in  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$ , respectively. We argue that no vertices  $(i')^{th}$  and  $(i' + 1)^{th}$  rows is outside  $\psi(R_{i_o}) \cup \psi(R_{i_o+1})$ . Note that  $R_{i_o}$  (similarly  $R_{i_o+1}$ ) is a separators in  $G$  such that there are at least two connected components of  $G - R_{i_o}$  (similarly  $G - R_{i_o+1}$ ) which has at least  $k+1$  vertices. By Observation 2.2,  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$  are two separators in  $\boxtimes_{r \times q}$ . If  $\psi(R_{i_o})$  or  $\psi(R_{i_o+1})$  are proper subset of vertices in  $(i_o)^{th}$  or  $(i_o+1)^{th}$  row then it can not be a separator in  $\boxtimes_{r \times q}$ . This implies  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$  correspond to two rows in  $\boxtimes_{r \times q}$ . Hence the number of columns in  $\boxtimes_{r \times q}$  is equal to  $|\psi(R_{i_o})| = |R_{i_o}| = q$ . ◀

We argue that if there is a large grid that is a separator in  $G$  then we can safely contract two consecutive rows in this grid.

▷ **Reduction Rule 7.2.** For an instance  $(G, k)$ , let  $S$  be a  $((4k+6) \times t)$ -grid-separator of  $G$  for some integer  $t$ . Let  $S_u (= \{u_1, u_2, \dots, u_t\})$  and  $S_v (= \{v_1, v_2, \dots, v_t\})$  be two consecutive internal rows in  $S$ . Let  $G'$  be the graph obtained from  $G$  by contracting all the edges in  $\{u_j v_j \mid j \in [q]\}$ . Return instance  $(G', k)$ .

We prove that the reduction rule is safe along the same line as that of Lemma 19.

▶ **Lemma 32.** *Reduction Rule 7.2 is safe.*

**Proof.** Note that contracting all edges across any two consecutive rows in a grid results in another grid.

( $\Rightarrow$ ) Assume  $G$  is  $k$ -contractible to  $\boxtimes_{r \times q}$  via mapping  $\psi$  for some positive integers  $r, q$ . By Lemma 31,  $q = t$ . We argue that  $G'$  is  $k$ -contractible to  $\boxtimes_{(r-1) \times q}$ . As argued in the proof of Lemma 31, there exists two consecutive rows  $R_{i_o}$  and  $R_{i_o+1}$  in  $G[S]$  such that  $\psi(R_{i_o})$  and  $\psi(R_{i_o+1})$  correspond to two rows in  $\boxtimes_{r \times q}$ . Since  $S_u, S_v$  are also rows in  $G[S]$ , we can conclude that  $\psi(S_u)$  and  $\psi(S_v)$  correspond to rows in  $\boxtimes_{r \times q}$ . Let  $\psi(S_u)$  and  $\psi(S_v)$  correspond to rows  $i_1, i_2$ . Since there are multiple edges across  $S_u, S_v$ , we have  $|i_1 - i_2| \leq 1$ . As  $G$  is  $k$ -contractible to  $\boxtimes_{r \times q}$ , if  $|i_1 - i_2| = 1$  then  $G'$  is  $k$ -contractible to  $\boxtimes_{(r-1) \times q}$ . If  $|i_1 - i_2| = 0$  then as  $G$  is  $k$ -contractible to  $\boxtimes_{r \times q}$  and  $G' = G / \{v_j u_j \mid j \in [q]\}$ ,  $G'$  is  $k$ -contractible to  $\boxtimes_{(r-1) \times q}$ .

( $\Leftarrow$ ) Let  $S_o = \{s_1, s_2, \dots, s_q\}$  be the set vertices in  $G'$  which are obtained by contracting edges  $u_j v_j$  in  $G$ . In other words, for  $j$  in  $[q]$ , let  $s_j$  be the new vertex added while contracting edge  $u_j v_j$ . Since  $S$  is a  $((4k+6) \times t)$ -grid-separator in  $G$ , set  $S' = (S \cup S_o) \setminus (S_u \cup S_v)$  is  $((4k+5) \times t)$ -grid-separator in  $G'$ .

Assume that  $G'$  is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$  via mapping  $\phi$ . By Lemma 31,  $q = t$ . We argue that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$ . By similar arguments as in previous part,  $\phi(S_o)$  corresponding to a row, say  $i_o$ , in  $\boxplus_{(r-1) \times q}$ . As  $G$  is  $q$ -contractible to  $G'$  (which is  $k$ -contractible to  $\boxplus_{(r-1) \times q}$ ), we know that  $G$  is  $(k+q)$ -contractible to  $\boxplus_{(r-1) \times q}$ . We define a mapping  $\psi : V(G) \rightarrow \boxplus_{r \times q}$  corresponding to this contraction as follows: For every  $x$  in  $V(G) \setminus (S_u \cup S_v)$ , define  $\psi(x) = \phi(x)$  and for every  $x$  in  $\{u_j, v_j\}$ , define  $\psi(x) = \phi(s_i)$ . We argue that  $i_o^{th}$  row in  $\boxplus_{(r-1) \times q}$  is partible. For every  $j$  in  $[q]$ , we define a partition  $U_j, V_j$  of  $\psi^{-1}([i_o, j])$  which satisfy all the properties mentioned in Definition 2. Since  $S$  is a  $(p \times t)$ -grid separator of  $G$ ,  $G - S$  has two connected component  $C_1, C_2$  as specified in Definition 29. Note that  $G - (S_u \cup S_v)$  also has exactly two connected components, say  $Y_1, Y_2$ , which contain  $C_1, C_2$ , respectively.

For  $j$  in  $[q]$ , let  $X_j = \psi^{-1}([i_o, j])$ . As  $s_j$  was present in  $\phi^{-1}([i_o, j])$ , vertices  $u_j, v_j$  are present in  $X_j$ . By the property of  $\phi$ , set  $\phi^{-1}([i_o, j])$  is connected in  $G'$ . Since vertex  $s_j$  is obtained from contracting edge  $u_j v_j$  in  $G$ , graph  $G[X_j]$  is connected. Since  $\phi(S_o)$  corresponds to a row of with  $q$  vertices and  $|S_o| = q$ , vertices  $u_{j'}, v_{j'}$  are present in  $X_j$  if and only if  $j' = j$ . In other words,  $X_j \cap (S_u \cup S_v) = \{u_j, v_j\}$ . Define  $U_j := (X_j \cap Y_1) \cup \{u_j\}$  and  $V_j := (X_j \cap Y_2) \cup \{v_j\}$ . Since  $(Y_1, S_u, S_v, Y_2)$  is a partition of  $V(G)$ , sets  $U_j, V_j$  is a non-empty partition of  $X_j$ . We argue that  $U_j, V_j$  satisfy all the properties in Definition 2.

As  $N(Y_1) = S_u$ , no vertex in  $U_j \setminus \{u_j\}$  is adjacent with  $S_v \cup Y_2$ . By similar arguments, no vertex in  $V_j \setminus \{v_j\}$  is adjacent with  $S_v \cup Y_1$ . As  $G[X_j]$  is connected and  $U_j \subseteq S_u \cup Y_1; V_j \subseteq S_v \cup Y_2$ , graphs  $G[U_j], G[V_j]$  are connected. Moreover, for  $j' \in [q]$ , sets  $U_j, V_{j'}$  are adjacent if and only if  $u_j, v_{j'}$  are adjacent. Since  $G[S_1 \cup S_2]$  is a  $(2 \times q)$ -grid,  $u_j$  and  $v_{j'}$  are adjacent if and only if  $j = j'$ . Hence  $U_j$  and  $V_{j'}$  are adjacent if and only if  $j = j'$ . Since  $U_j \subseteq X_j$  and  $U_{j'} \subseteq X_{j'}$ ,  $U_j$  and  $U_{j'}$  are non adjacent if  $|j - j'| > 1$ . If  $|j - j'| = 1$  then  $U_j, U_{j'}$  are adjacent as they contain  $u_j$  and  $u_{j'}$ . Hence  $U_j, U_{j'}$  are adjacent if and only  $|j - j'| = 1$ . By similar arguments,  $V_j, V_{j'}$  are adjacent if and only if  $|j - j'| = 1$ . As no vertex in  $U_j$  is adjacent with  $Y_2$  and no vertex in  $V_j$  is adjacent with  $Y_1$ , we can conclude that partition  $U_j, V_j$  satisfy all the properties in Definition 2. Since  $i_o^{th}$  row in  $\boxplus_{(r-1) \times q}$  is partible, Lemma 3 implies that  $G$  is  $k$ -contractible to  $\boxplus_{r \times q}$ . This concludes the proof of reverse direction.  $\blacktriangleleft$

The following lemma, which is analogous to Lemma 21, is essential to argue that Reduction Rule 7.2 can be applied in polynomial time.

► **Lemma 33.** *There exists an algorithm which given an instance  $(G, k)$  of GRID CONTRACTION and integers  $p, t$  runs in polynomial time and either returns a  $(p \times t)$ -grid-separator of  $G$  or correctly concludes that no such separator exists.*

**Proof.** The algorithm guesses the two vertices in the first column which are in first and last row of a potential  $(p \times t)$ -grid-separator of the graph. It considers all pairs of vertices  $u_1, u_p$  in  $G$  which are at distance  $p$  from each other and there is a unique shortest path between  $u_1$  and  $u_p$ . Let  $(u_1, u_2, \dots, u_p)$  be the unique shorted path. For  $i \in \{1, 3, \dots, p-1\}$ , the algorithm tries to find a subset  $S_i$  of  $V(G)$  which has following properties: (a)  $G[S_i]$  is a  $(2 \times t)$  grid, (b)  $u_i, u_{i+1}$  are two vertices in the first column of  $G[S_i]$ , and (c) each row in  $S_i$  is a separator in  $G$ . If such subset exists for every  $i$  then the algorithm checks if  $G - S$ , where  $S = \bigcup_i S_i$ , has two connected component each with at least  $k+1$  vertices. If it is the case then the algorithm returns  $S$ . If not it moves to the next pair of vertices. If it does not find such a set for any pair of vertices, it concludes that the graph does not contain a  $(p \times t)$ -grid-separator.

The algorithm returns a  $(p \times t)$ -grid separators only it had found one. Now, suppose that the graph has  $(p \times t)$ -grid separator  $S$ . Let  $G[S] = \boxplus_{p \times t}$ ,  $u_1 = [1, 1]$ , and  $u_p = [p, 1]$ . The

vertices in the first row form a unique shortest path of length  $p$  between  $u_1, u_p$ . Moreover, every consecutive two rows in  $G[S]$  satisfies the three conditions mentioned in the above paragraph. Hence, if the graph has a  $(p \times t)$ -grid separator then the algorithm returns it. It remains to argue the running time of the algorithm. The algorithm runs over all pairs of vertices which are at distance  $p$ . It can find all such pairs exhaustively within polynomial time. The algorithm then uses Lemma 20 to find the desired sets for  $p$  pair of vertices. The running time of the overall algorithm is implied by Lemma 20 and the fact that all other steps in it can be executed in polynomial time.  $\blacktriangleleft$

We are now in a position to present the main result of the section.

► **Theorem 34.** GRID CONTRACTION admits a kernel with  $\mathcal{O}(k^4)$  vertices and edges.

**Proof.** Recall that  $k_0 = (4k + 8)(k + 1) + 1$ . We assume that the input graph is connected as otherwise it is safe to conclude that we are working with a NO instance. Given an instance of GRID CONTRACTION, the kernelization algorithm exhaustively applies Reduction Rule 7.1. Let  $(G, k)$  be the resultant instance. If  $k \leq 0$  and  $G$  is not a grid then the algorithm returns a NO instance. If the number of vertices in the resulting instance is at most  $k_0^2 + k + 1$  then the algorithm returns it as a kernel. Consider a case when the number of vertices in the resulting instance is more than  $k_0^2 + k + 1$ . The algorithm uses Lemma 33 to find the largest value of  $t$  that is smaller than  $|V(G)|$  and there exists a  $((4k + 6) \times t)$ -grid-separator in  $G$ . If no such  $t$  exists then the algorithm returns a NO instance. It then applies Reduction Rule 7.2 to obtain a smaller instance. The algorithm repeats the process until the number of vertices in the reduced graph is at most  $k_0^2 + k + 1$  or it can not find a  $((4k + 6) \times t)$ -grid-separator. In the first case, it returns the reduced instance as a kernel while in another case it returns a NO instance.

By Lemma 30, if  $(G, K)$  is a YES instance there exists a  $((4k + 6) \times t)$ -grid-separator in  $G$  for some integer  $t$ . This, along with Lemma 28 and 19 imply the correctness of the algorithm. The algorithms run in polynomial time by Lemma 33 and the fact that each application of Reduction Rule reduces the number of vertices by  $t$ . As Reduction Rule 7.2 is not applicable, the reduced graph has  $\mathcal{O}(k^4)$  vertices. Since Reduction Rule 7.1 does not apply to the reduced instance, there are at most  $6k$  vertices whose degree is more than 5 and less than  $k + 5$ . The number of edges incident on these vertices is  $\mathcal{O}(k^2)$ . As remaining vertices have a degree at most 4, the number of edges in the reduced graph is  $\mathcal{O}(k^4)$ . This implies the reduced graph has the desired number of vertices and edges.  $\blacktriangleleft$

## 8 Conclusion

In this article, we study the parameterized complexity of GRID CONTRACTION. We present an FPT algorithm running that given an instance  $(G, k)$  of the problem runs in time  $4^{6k} \cdot n^{\mathcal{O}(1)}$  and correctly determines whether it is a YES instance. We present a notation of  $r$ -slab which is a generalization of a connected component of a graph. We believe this or similar notation might be useful to get FPT or Exact Exponential Algorithms. We prove that unless ETH fails, there is no algorithm for GRID CONTRACTION that runs in time  $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ . Finally, we prove that problem admits a kernel with  $\mathcal{O}(k^4)$  vertices and edges.

To the best of our knowledge, GRID CONTRACTION is the only problem that admits a polynomial kernel when target graph class has unbounded path-width. It is an interesting question to find another graph class  $\mathcal{G}$  such that  $\mathcal{G}$  has an unbounded width-parameter and  $\mathcal{G}$ -CONTRACTION admits a polynomial kernel.

## References

- 1 Akanksha Agrawal, Fedor Fomin, Daniel Lokshtanov, Saket Saurabh, and Prafullkumar Tale. Path contraction faster than  $2^n$ . *The 46th International Colloquium on Automata, Languages and Programming (ICALP 2019)*, 2019.
- 2 Akanksha Agrawal, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. Split contraction: The untold story. *ACM Transactions on Computation Theory (TOCT)*, 11(3):1–22, 2019.
- 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 Andries Evert Brouwer and Henk Jan Veldman. Contractibility and NP-completeness. *Journal of Graph Theory*, 11(1):71–79, 1987.
- 6 Leizhen Cai and Chengwei Guo. Contracting few edges to remove forbidden induced subgraphs. In *International Symposium on Parameterized and Exact Computation*, pages 97–109. Springer, 2013.
- 7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 8 Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized complexity*. Springer-Verlag, 2013.
- 9 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006.
- 10 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.
- 11 Petr A Golovach, Marcin Kamiński, Daniël Paulusma, and Dimitrios M Thilikos. Increasing the minimum degree of a graph by contractions. *Theoretical computer science*, 481:74–84, 2013.
- 12 Petr A. Golovach, Pim van ’t Hof, and Daniel Paulusma. Obtaining planarity by contracting few edges. *Theoretical Computer Science*, 476:38–46, 2013.
- 13 Sylvain Guillemot and Dániel Marx. A faster FPT algorithm for bipartite contraction. *Inf. Process. Lett.*, 113(22–24):906–912, 2013.
- 14 Pinar Heggeres, 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.
- 15 Pinar Heggeres, Pim Van’t Hof, Benjamin Lévêque, Daniel Lokshtanov, and Christophe Paul. Contracting graphs to paths and trees. *Algorithmica*, 68(1):109–132, 2014.
- 16 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512 – 530, 2001.
- 17 R. Krithika, Pranabendu Misra, Ashutosh Rai, and Prafullkumar Tale. Lossy kernels for graph contraction problems. In *36th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2016*, pages 23:1–23:14, 2016.
- 18 R Krithika, Pranabendu Misra, and Prafullkumar Tale. An FPT algorithm for contraction to cactus. In *International Computing and Combinatorics Conference*, pages 341–352. Springer, 2018.
- 19 Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. On the hardness of eliminating small induced subgraphs by contracting edges. In *International Symposium on Parameterized and Exact Computation*, pages 243–254, 2013.
- 20 Rolf Niedermeier. *Invitation to fixed-parameter algorithms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.
- 21 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.



- 22 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.